

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

THIS PAGE BLANK (USPTO)



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **06259426 A**(43) Date of publication of application: **16.09.94**

(51) Int. Cl.

G06F 15/20(21) Application number: **05049042**(22) Date of filing: **10.03.93**(71) Applicant: **CANON INC**

(72) Inventor:

YAMAGUCHI HIROSHIGE
SAKAI TETSUO
ISHIGURO KENJI
KASHIWAGI MASAKI
MATSUKI HIROSHI
SASAKI YASUHIKO
OSADA MAMORU

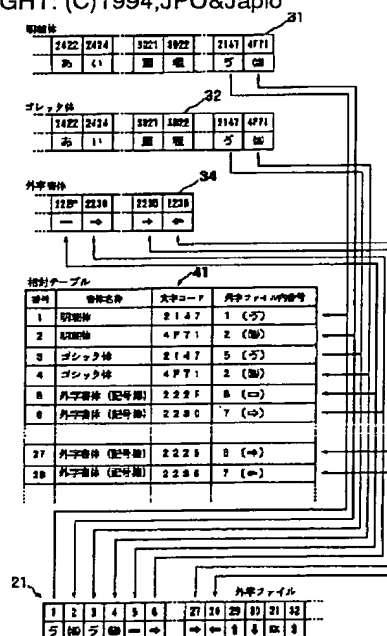
(54) METHOD AND DEVICE FOR PROCESSING CHARACTER**(57) Abstract:**

PURPOSE: To constitute the device so that an external character in which a conception of a style of handwriting exists can be managed by allocating the same code to the same kind of external character between different styles of handwriting, and the compatibility of the code of the external character between the styles of handwriting.

CONSTITUTION: In a relative table 41, the name of a style of handwriting, and a character code are registered in advance with respect to an in-external character file number for specifying an external character. Accordingly, by the relative table 41, each external character in an external character file 21 is managed by being subordinated to each style-of-writing file (a Ming-style file 31, a Gothic style file 32, an external character style-of-handwriting file 34). At the time of obtaining an external character, external character data is obtained from the external character file 21 by designating the name of the style of handwriting and the character code. For instance, [U] of the Ming-style and [U] of the Gothic style are the same

kind of external characters, and the same character code [2147] is allocated.

COPYRIGHT: (C)1994,JPO&Japio



THIS PAGE BLANK (USPTO)

6-259426

[ABSTRACT]

[OBJECT]

To enable management of various kinds of external characters each having a type-face by assigning the same code to the same kind of external characters having different type-faces and to ensure conformity of external character codes between external characters having different type-faces.

[CONSTRUCTION]

The correlation table 41 shows that the names of face types and codes of the characters are registered with respect to the numbers in the external character file for specifying the external characters. Accordingly, each external character in the external character file 21 is managed by the correlation table 41 subordinate to each face type file (the mincho face type file 31, the bold face type file 32 and the external character face type file 34). When external characters are desired, external character data are acquired from the external character file 21 by designating the face type name and character code. For example, u" in the mincho face type and u" in the bold face type are the same type of external character, and the same character code "2147" is allotted to them.

- a Mincho face type
- b Bold face type
- c External character face type
- d Correlation table
- e Number
- f Name of face type
- g Character code
- h Number in external character file

THIS PAGE BLANK (USPTO)

- i Mincho face type
- j Bold face type
- k External character face type (Marks)
- l External character file

[0012]

EMBODIMENT 1

Fig. 1 is a block diagram showing a basic arrangement of a character processing device of an embodiment the invention. This device may, for example, be a word processor. Alternatively, it may be a personal computer, a work station or a computer system.

[0013]

In Fig. 1, reference numeral 1 is a central processing unit (CPU) which controls the entire device and conducts arithmetic operations. Reference numeral 2 is a read-only memory (ROM) for storing a system-start program, character pattern data and character code information. Usually, the character pattern stored in ROM 2 is utilized as a system font for displaying characters of a menu. The character code information is information of the code system of the system font. This character code information may include information of new JIS and old JIS. In this connection, the system font may be face type data on ROM 2 or face type previously designated by a user. Reference numeral 3 is a random access memory (RAM) which is a data storing region on which no restrictions are placed when it is used. In other words, various programs and data are loaded in RAM 3 when various processing is conducted.

[0014]

Reference numeral 4 is a keyboard control section (KBC) which receives input data from the keyboard 5 and transmits the data to CPU 1. Reference numeral 6 is a display control section (CRTC) which outputs display data to CRT 7. Reference numeral 7 is a display device (CRT) which receives data from CRTC 6 and displays the received data on it.

THIS PAGE BLANK (USPTO)

Reference numeral 8 is a disk control section (DKC) which transmits data to an external storing device. Reference numeral 9 is an external storing device provided with a floppy disk device (FD) or hard disk device (HD). The external storing device 9 stores programs and data. These programs and data are referred to, if necessary, when various processing is carried out by CPU 1, or loaded into RAM 3. For example, when the operation system demands a necessary processing, the system or program for conducting the processing is loaded into RAM 3 from the external storing device 9 and started by CPU 1. While the system or program is being carried out, CPU 1 loads data and information, which are stored in the external storing device 9, on RAM 3, if necessary. A control program expressed by a flow chart described later is also loaded into RAM 3 and carried out by CPU 1.

[0015]

Reference numeral 10 is a printer control section (PRTC) which outputs recorded data into a printer 11, which forms a visible image on a recording medium according to recorded data received from PRTC 10. Reference numeral 12 is a system bus which functions as a data passage between the above components.

[0016]

Fig. 2 is a view showing the construction of data in RAM 3. Reference numeral 201 is a window system region in which the window system program is loaded. Reference numeral 202 is a window system work region which is used by the window system when various processing is carried out. Reference numeral 203 is a management region which stores management information with respect to various device drivers, such as a printer driver and CRT driver, and also stores management information with respect to a font driver. Reference numeral 204 is a font management information table region for storing a font management information table to register information relating to a font, such as the face type and system font handled by each font driver. Reference numeral 205 is a

THIS PAGE BLANK (USPTO)

window system cache region of character data managed by the window system. Reference numeral 206 is a font driver region in which the font driver used by the window system is stored. Reference numeral 207 is a correlation table region in which the correlation table, described later, is stored.

[0017]

Fig. 3 is a block diagram showing the basic structure of the window system. Reference numeral 301 is a window system in which OS may be included. The window system 301 is loaded into the window system region 201 of RAM 3 and started by CPU 1. Reference numeral 302 is a font management information table which contains information with respect to the face type which are possessed by the font driver that is controlled by the window system 301, and the font management information table also has information with respect to characters the acquirement of which is demanded by an application. The font management information table 302 is stored in the font management information table region 204 of RAM 3. Reference numerals 303 and 304 are applications started by the window system 301. The applications 303 and 304 are respectively defined as application A and application B.

[0018]

Various device drivers 305 to 309 are programs for controlling the devices composing the present system. Printer driver A 305 and printer driver B 306 are programs for controlling the printer. CRT driver 307 is a program for controlling a CRT, the keyboard driver 308 is a program for controlling the keyboard, and the mouse driver 309 is a program for controlling the mouse. These various drivers 305 to 309 are loaded into RAM 3 and managed by the window system 301 according to the management information stored in the management region 203.

[0019]

Reference numerals 310 and 311 are respectively font drivers A and B, which are means for making and providing demanded characters. The font drivers A and B may be

THIS PAGE BLANK (USPTO)

provided as a device or program. In the case where the font drivers A and B are provided as a program, for example, they are carried out after they have been loaded into RAM 3. The font drivers 310 and 311 may be directly managed by the window system 301. Alternatively, the font drivers 310 and 311 may be managed by various device drivers such as printer drivers 305, 306 or a display driver 307, or included in the drivers.

[0020]

Figs. 4 and 5 are flow charts showing a procedure for starting the window system and also showing a procedure for acquiring characters demanded by the application.

[0021]

In step S11, the window system 301 is loaded in the window system region 201 of RAM 3, and the thus loaded window system 301 is started by CPU 1. At this time, the window system work region 202 is reserved on RAM 3, and the management region 203 is reserved according to the number and state of various device drivers and font drivers controlled by the window system. Further, the font management information table region 204 and the window cache region 205 are reserved on RAM 3, and their contents are initialized.

[0022]

Next, in step S12, the font drivers (310 to 311) are loaded in the font driver region 206 of RAM 3 and then started. In step S13, an initial image plane of the window system is displayed on CRT 7. In step S14, face type information is required to acquire the font drivers (310 to 311). In step S15, an information table (a portion of the font management information table 302) is made according to the information provided by the font drivers (310 to 311) in step S14 and stored in the font management information region 204 of RAM 3. As described above, the window system completes the initial motion including a display of the initial image plane. In step S16, the device waits for various events sent from the input device 5 such as a keyboard and mouse, i.e. the device waits for an event for

THIS PAGE BLANK (USPTO)

starting the application.

[0023]

When the applications 303 and 304 (the number may be one or plural) are started, the program proceeds to step S17, and the device waits for events generating a demand to acquire characters or a demand to set type-faces or sizes of characters. When an event of an application is generated, the program proceeds to step S18. Processing after step S18 is conducted according to the occurrence of the event of the application. In this case, processing of only one (application A303 in this case) of a plurality of applications, which are managed by the window system, will be explained here. In this connection, when the plurality of applications are started, processing after step S18 proceeds independently with respect to each application, and the window system manages each application.

[0024]

In step S18, it is judged whether or not the event, which is generated to execute a process in the application A303 is a demand to acquire face type information to the font drivers 310 and 311 (The number of the font drivers may be one or plural.). In this case, when the event is the demand to acquire face type information, the program proceeds to step S22. In step S22, the window system 301 issues a demand to acquire information of all face types, which are owned by the font driver, to all font drivers 310, 311 managed by the present window system 301. In response to the information demand from the window system, each font driver transmits all face type information (face type information), which are possessed by the font driver, to the window system. After the window system has received the face type information from each font driver, the window system transmits all the face type information to the application A303. This series of processing is conducted to transmit the face type information to the application in step S22. After processing has been carried out in step S22, the program returns to step S17 in which the device waits for an event to execute the

THIS PAGE BLANK (USPTO)

application.

[0025]

In the case where the event generated in step S18 is not a demand for acquiring face type information, the program proceeds to step S19 which is the next processing.

[0026]

In step S19, it is judged whether or not the event is a demand of acquiring characters (demand of acquiring characters to the window system). In the case where the event is not a demand of acquiring characters, the program proceeds to step S23, and various processing corresponding to the event is carried out and then the program returns to step S17 in which the device waits for an event for generating the application. In the case where the event of demand to acquire characters (offering characters to the window system) is generated in step S19, the program proceeds to step S20 which is the next processing, and the process to acquire characters is carried out.

[0027]

In step S20, it is judged whether or not character data (demand of acquiring various information including size designation and face type selection) demanded by the application A303 to be acquired are stored in the window system cache region 205 managed by the window system 301. This judgement is carried out by retrieving the font management information table 302. In the case where the character data are stored in the cache region, the program proceeds to step S21, and the window system 301 sends the characters stored in the window system cache region 205 to CRT driver 307 and demands to obtain an output. On the other hand, when the designated characters are not stored in the window system cache region 205, the program proceeds from step S20 to step S24.

[0028]

In step S24, according to the name and the size of the designated face type, it is decided to which font driver a demand of outputting characters is made. In step S25,

THIS PAGE BLANK (USPTO)

information from the font driver, which has made a demand of outputting the characters by the unit of "face type" and "size" demanded by the application, is registered on the font information management table 302. The font information management table 302 manages the characters demanded by all the applications managed by the window system by the unit of "face type" and "size". In step S26, a demand of outputting the characters is made on the font driver decided in step S24. At this time, the font driver to which the demand of outputting the characters is made offers the demanded characters to the window system 301. In step S27, the character pattern offered by the font driver in step S26 is stored in the window system cache region 205 managed by the window system 301.

[0029]

After that, the program proceeds to step S21, and the characters are outputted to the display, and then the program returns to step S17 in which the device waits for an event of the application. This processing from step S17 to step S21 is continued until the application is completed.

[0030]

Next, referring to the flow chart shown in Fig. 6, a basic motion for acquiring the characters of the application started by the window system 301 will be explained below. As an example, a case in which the application A303 is started will be explained here.

[0031]

When an event of making a demand of starting the application A303 is issued by the window system 301, an initial image plane is displayed in step S31. A user is notified by this initial image plane that the application is put into a state in which it can be used. After that, in order to acquire all information of the usable face type at present, the application A303 makes an inquiry via the window system 301 to the drivers (printer drivers 305, 306, display driver 307 and so forth) of all the output devices managed by the window system. Then, the application A303 makes and

THIS PAGE BLANK (USPTO)

displays a face type list (shown in Fig. 7) in which the thus obtained usable face type names are listed.

[0032]

In step S33, the user is made to select a desired face type from the face type list (shown in Fig. 7). Also, the user is made to set all parameters with respect to the characters. In this case, the parameters are the size of the characters, the pitch of the characters, the line and row pitch, the modification of the characters, and the thickness of the characters, which are designated by the user.

[0033]

In step S34, characters such as "hiragana", "katakana", "kanji", alphameric characters and marks are inputted through the keyboard 5, and then the program proceeds to step S35. In step S35, in order to acquire characters inputted in step S34 according to the face type and parameter which have been set in step S33, a demand of outputting the characters is made to the window system 301. What is called processing to acquire (output) characters is described from step S34 to step S35. While the application is being carried out, this processing is usually repeated.

[0034]

Next, an outline of the method of managing characters, which is conducted in the present embodiment 1, will be explained below. In the present embodiment 1, a case will be explained as an example in which the external character file (shown in Fig. 8) to accommodate a pattern (referred to as external characters hereinafter) originally made by a user is stored together with a plurality of usual face type files (shown in Fig. 9). In this embodiment 1, the correlation table (shown in Fig. 10) is used, by which each character in the external character file (shown in Fig. 8) can be related to the usual face type file (shown in Fig. 9). That is, the managing method of managing characters and face types of the embodiment 1 is defined as a method of managing while each external character in the external character file 21 is subordinated to a usual face type by using the correlation

THIS PAGE BLANK (USPTO)

table.

[0035]

Fig. 8 is a view for explaining the content of the external character file for accommodating characters (external characters) made by the user. Reference numeral 21 represents the entire external character file. A number attached to the upper row of the external character file 21 is an external character managing number 21a which has no relation with the character code. That is, this number attached to the upper row of the external character file 21 is peculiar to this external character file 21. Corresponding to the external character managing number 21a, the external character patterns 21b are registered one by one.

[0036]

Fig. 9 is a view showing a data structure of the usual face type file. Corresponding to the character codes 31a, character data 31b are accommodated. Concerning the face type files, there are provided a mincho face type file 31, bold face type file 32 and mohitsu face type file 33.

[0037]

Fig. 10 is a view showing a data structure of the correlation table by which each external character in the external character file is related to the usual face type file. The correlation table 41 is used for managing to what face type each external character, which is registered in the external character file 21, is subordinate, and also the correlation table 41 is used for managing to what character code each external character is made to correspond. In Fig. 10, reference numeral 41a is a sequence number representing the order on the correlation table. Reference numeral 41d is a number used in the external character file, which corresponds to the external character managing number 21a in the external character file 21. Reference numeral 41b is a face type name, which is information for discriminating the face type to which the external character indicated by the number 41d in the external character file is subordinate.

THIS PAGE BLANK (USPTO)

Reference numeral 41c is a character code, in which a character code allotted to the external character indicated by the number 41d in the external file is registered.

[0038]

For example, in the external character file 21 shown in Fig. 8, under the condition that "u", the external character managing number 21a of which is 1, is of mincho face type, "u", the external character managing number 21a of which is 5, is of a bold face type, and "u", the external character managing number 21a of which is 16, is of mohitsu face type, the character code is allotted to "2147" of JIS code. In this case, information about "u" of mincho face type is registered as shown in the first column on the correlation table 41. In the same manner, "u" of the bold face type, the external character managing number of which is 5, is registered in the fifth column on the correlation table 41, and "u" of mohitsu face type, the external character managing number of which is 16, is registered in the sixteenth column on the correlation table 41.

[0039]

In the character processing device of this embodiment 1, for example, in the case where a demand of acquiring a character, the bold face type character code of which is "2147", is issued, it is retrieved whether or not a character of the face type concerned exists on the correlation table 41. That is, an external character, the face type name 41b of which is "the bold face type", and the character code 41c of which is "2147", is retrieved. In the case where the external character concerned exists on the correlation table, the character information is extracted from the external character file 21 and developed into a character pattern. In this way, the character "u" of the bold face type is acquired.

[0040]

Concerning the external characters, which are shown in the external character file 21 by the number from 6 to 15, which are not accompanied by the concept of the face type so

THIS PAGE BLANK (USPTO)

that they should be processed as marks or figures, the external characters are defined as "external face type" which is a singly independent face type so that a user can discriminate the face type and manage them.

[0041]

Fig. 11 is a conceptual view for simply expressing a relation among the external character file 21 shown in Fig. 8, the usual face type file 31 shown in Fig. 9 and the correlation table 41 shown in Fig. 10. The essential concept of the character processing device of the embodiment 1 is to realize the management having the above relevance. That is, realizing the management having the above relation is the managing method itself.

[0042]

Next, further explanations will be made into registration of the external characters and processing of acquirement of the external characters in the character processing device of embodiment 1 having the above structure.

[0043]

First, referring to the flow chart shown in Fig. 12, processing of registering an external character, which is made by a user, will be explained below. Fig. 12 is a flow chart showing a procedure of processing of registration conducted in the character processing device of embodiment 1. The steps of this flow chart are carried out when a demand of registration of an external character is issued to the character processing device after the user has made the external character. In this connection, in the demand of registration, the external character to be registered, the face type and the character code are designated.

[0044]

First, in step S41, it is judged whether or not one or more external characters have already been registered. In the case where the registered external characters do not exist, the program proceeds to step S42. In step S42, a region in which the correlation table 41 for managing the external characters is ensured on RAM 3, and at the same time

THIS PAGE BLANK (USPTO)

the device is set at the initial state. Then, the program proceeds to step S45.

[0045]

On the other hand, in the case where some external characters have already been registered in step S41, the program proceeds to step S43. In step S43, retrieval is conducted on the correlation table 41 according to the designated face type and character code. In step S44, according to the result of retrieval conducted in step S43, it is judged whether or not the external characters concerned already exist. In the case of non-existence, the program proceeds to step S45. In the case of existence, the program proceeds to step S47.

[0046]

In step S45, external characters, which have been made, are newly registered. In this case, the external characters to be registered are added to the end of the external character file 21. In this case, the external characters, which have been made, are added to the end of the external character file 21. However, it should be noted that the present invention is not limited to the above specific embodiment. For example, the external characters, which have been made, may be added to the head of the external character file 21. Alternatively, the external characters, which have been made, may be inserted at an appropriate position according to the frequency of appearance or the order of character codes. In step S46, information such as a face types and character code of the external character registered in step S45 is registered on the correlation table 41. In this way, processing is completed.

[0047]

On the other hand, in the case where the external characters concerned exist in step S44, the program proceeds to step S47. Processing after step S47 corresponds to a renewal processing for the external characters which have been made and registered before. In step S47, the external characters concerned in the external character file 21 are

THIS PAGE BLANK (USPTO)

deleted, and the newly made external characters are registered. In step S48, the portion concerned on the correlation table 41 is renewed by the information (face types and character codes) about the newly registered external characters. That is, the relative information about the external characters present in the portion concerned on the correlation table is deleted, and the relative information about the newly registered external characters is registered in the portion.

[0048]

In this connection, although not described in the flow chart of Fig. 12, the following are clear. In the case where the external characters are deleted, concerning the designated external characters, the characters can be deleted from the external character file 21 by the same procedure, and at the same time the corresponding relative information can be deleted from the correlation table 41 by the same procedure.

[0049]

When the above processing is repeated, it is possible to grasp and manage the latest state at all times when external characters are newly made, renewed, changed, corrected and deleted.

[0050]

Next, referring to the flow chart of Fig. 13, a procedure carried out in the case of acquiring character images will be explained below. Fig. 13 is a flow chart expressing a procedure for generating a character pattern in the character processing device of the embodiment 1. In this connection, this processing is carried out at a point of time when a demand of acquiring a character pattern is issued.

[0051]

In step S51, retrieval is conducted on the correlation table 41 according to the designated face type and character code. In step S52, it is judged whether or not the designated character is an external character according to the result of retrieval conducted in step S51. That is, when

THIS PAGE BLANK (USPTO)

the designated character is present on the correlation table 41, it is judged to be an external character, and when the designated character is not present on the correlation table 41, it is judged to be a usual character. When it is judged in step S52 that the character is an external character, the program proceeds to step S54. When it is judged in step S52 that the character is not an external character, the program proceeds to step S53.

[0052]

In step S53, character data are acquired from the usual face type file, and the character pattern is generated and outputted.

[0053]

In step S54, according to the result of retrieval conducted on the correlation table 41 in step S51, an accommodating position is specified in the external character file 21 of the pattern data of the external characters having the designated face type and character code. In step S55, pattern data of the external character concerned are acquired from the external character file 21, and the character pattern is generated and outputted.

[0054]

As described above, when data of the external character file 21, usual face type file 31 and correlation table 41 are used, it is possible to manage while discriminating between the external characters, which are subordinate to the usual face type, and the external character face type which is a group of only external characters.

[0055]

As described above, according to embodiment 1, external characters to be processed as characters are subordinate to the usual face type for each face type, and only external characters such as simple illustrations or marks, which are not processed as characters, can be processed as an independent single face type. Concerning the external characters to be processed as characters, the same character code is given between the face types. Therefore, it is

THIS PAGE BLANK (USPTO)

possible to maintain conformity of external characters between the face types. Further, a relation between the character pattern to be acquired and the character code can be managed in the character processing device without relying on the memory of the user. Therefore, system flexibility is enhanced.

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁(JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 6 - 2 5 9 4 2 6

(43) 公開日 平成6年(1994)9月16日

(51) Int. Cl. ⁵

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 15/20

5 6 2 M 7315-5 L

N 7315-5 L

審査請求 未請求 請求項の数 1 0

O L

(全 3 1 頁)

(21) 出願番号 特願平5-49042
(22) 出願日 平成5年(1993)3月10日

(71) 出願人 000001007
キヤノン株式会社
東京都大田区下丸子3丁目30番2号
(72) 発明者 山口 裕成
東京都大田区下丸子3丁目30番2号 キヤノ
ン株式会社内
(72) 発明者 酒井 哲夫
東京都大田区下丸子3丁目30番2号 キヤノ
ン株式会社内
(72) 発明者 石黒 健二
東京都大田区下丸子3丁目30番2号 キヤノ
ン株式会社内
(74) 代理人 弁理士 大塚 康徳 (外1名)

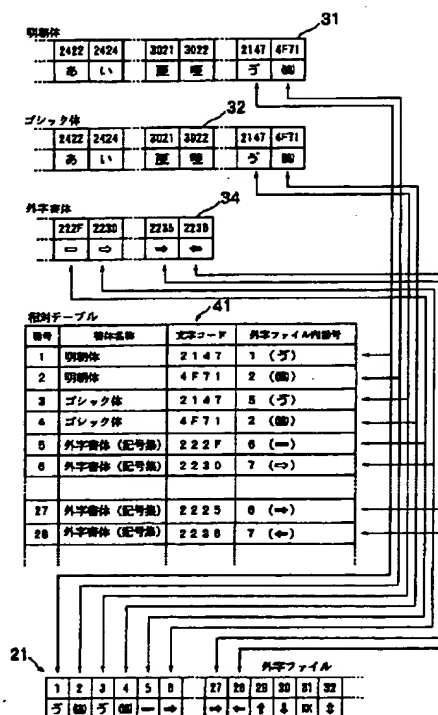
最終頁に続く

(54) 【発明の名称】 文字処理方法及び装置

(57) 【要約】

【目的】 書体の概念が存在する外字について、異なる書体の間の同種の外字に対して同一のコードを割り付けて管理することを可能とし、書体間における外字のコードの整合性を保つ。

【構成】 相対テーブル 4 1 には、外字を特定する外字ファイル内番号に対して書体名称、文字コードが登録されている。従って、相対テーブル 4 1 により、外字ファイル 2 1 内の各外字は各書体ファイル (明朝体ファイル 3 1、ゴシック体ファイル 3 2、外字書体ファイル 3 4) に従属して管理される。外字取得に際しては、書体名称及び文字コードを指定することにより、外字ファイル 2 1 より外字データを取得する。例えば、明朝体の「うゝ」とゴシック体の「うゝ」は同種の外字であり、同一の文字コード「2 1 4 7」が割り当てられている。



【特許請求の範囲】

【請求項1】 ユーザがデザインした外字の登録及び利用が可能な文字処理装置であって、登録された外字と該外字が所属する書体とを対応づける対応情報を登録する登録手段と、前記所属する書体を異にする同種の外字に対して同一の識別コードを付与するコード付与手段と、前記所属する書体及び前記識別コードを用いて所望の外字を取得する取得手段と、を備えることを特徴とする文字処理装置。

【請求項2】 前記登録手段により登録される対応情報、及び前記コード付与手段に付与される識別コードを記憶するための情報テーブルを更に備え、前記取得手段は、前記所属する書体及び前記識別コードにより前記情報テーブルを検索することにより所望の文字を取得する、ことを特徴とする請求項1に記載の文字処理装置。

【請求項3】 書体もしくは前記識別コードを単位として前記情報テーブルを検索するためのインデックステーブルを更に備え、前記取得手段は、前記所属する書体及び前記識別コードにより前記インデックステーブルを用いて前記情報テーブルを検索することにより所望の文字を取得する、ことを特徴とする請求項2に記載の文字処理装置。

【請求項4】 前記取得手段において前記所属する書体及び前記識別コードで指定された外字が未登録のとき、他の書体で同一識別コードを有する外字をもって代用する代用外字取得手段を更に備えることを特徴とする請求項1乃至3のいずれかに記載の文字処理装置。

【請求項5】 ユーザがデザインした外字をベクトルデータで登録し、これを展開して利用することが可能な文字処理装置であって、登録された外字と該外字が所属する書体とを対応づける対応情報を登録する登録手段と、前記所属する書体を異にする同種の外字に対して同一の識別コードを付与するコード付与手段と、前記所属する書体及び前記識別コードを用いて所望の外字のベクトルデータを取得する取得手段と、前記所属する書体に基づいてヒント情報の適用を変更して前記取得手段で取得されたベクトルデータの展開を行う展開手段と、を備えることを特徴とする文字処理装置。

【請求項6】 ユーザがデザインした外字の登録及び利用が可能な文字処理方法であって、登録された外字と該外字が所属する書体とを対応づける対応情報を登録する登録工程と、前記所属する書体を異にする同種の外字に対して同一の識別コードを付与するコード付与工程と、前記所属する書体及び前記識別コードを用いて所望の外字を取得する取得工程と、

を備えることを特徴とする文字処理方法。

【請求項7】 前記登録工程により登録される対応情報、及び前記コード付与工程に付与される識別コードを記憶するための情報テーブルを更に備え、前記取得工程は、前記所属する書体及び前記識別コードにより前記情報テーブルを検索することにより所望の文字を取得する、ことを特徴とする請求項6に記載の文字処理方法。

【請求項8】 書体もしくは前記識別コードを単位として前記情報テーブルを検索するためのインデックステーブルを更に備え、前記取得工程は、前記所属する書体及び前記識別コードにより前記インデックステーブルを用いて前記情報テーブルを検索することにより所望の文字を取得する、ことを特徴とする請求項7に記載の文字処理方法。

【請求項9】 前記取得工程において前記所属する書体及び前記識別コードで指定された外字が未登録のとき、他の書体で同一識別コードを有する外字をもって代用する代用外字取得工程を更に備えることを特徴とする請求項6乃至8のいずれかに記載の文字処理方法。

【請求項10】 ユーザがデザインした外字をベクトルデータで登録し、これを展開して利用することが可能な文字処理方法であって、登録された外字と該外字が所属する書体とを対応づける対応情報を登録する登録工程と、前記所属する書体を異にする同種の外字に対して同一の識別コードを付与するコード付与工程と、前記所属する書体及び前記識別コードを用いて所望の外字のベクトルデータを取得する取得工程と、前記所属する書体に基づいてヒント情報の適用を変更して前記取得工程で取得されたベクトルデータの展開を行う展開工程と、を備えることを特徴とする文字処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は文字や記号等のパターン出力を行う文字処理方法及び装置に関し、特にユーザがデザインした外字の登録及び利用が可能な文字処理方法及び装置に関する。

【0002】

【従来の技術】 従来、パーソナルコンピュータ（以下パソコンと称する）、ワードプロセッサ（以下ワープロと称する）、ディスクトップパブリッシング装置（以下DTP装置と称する）等における文字処理装置は、使用者が独自に作成した文字パターンやイメージパターン（以降これらを外字と称する）を登録するとともに、これらが必要に応じて呼び出して利用することができるという、外字機能を有している。外字の取り扱いが可能な従来の文字処理装置においては、人名や地名等に使用される文字パターンと、記号、シンボル、干支等のイラストに類

するイメージパターンとが、区別なく一律に「外字」として管理されている。

【0003】例えば、「ウ」に濁点を付けた「ヴ」という文字は、JISによる規格ではカタカナにのみ存在し、ひらがなには存在しない。従ってひらがなの「ヴ（う゛）」を使う場合には、独自に外字として作成する必要がある。そして従来の文字処理装置においては、「う゛」のように文字として扱うべき外字と、簡単なイラストのようにイメージパターンとして扱うべき外字とを、区別なく一律に外字、又は外字集、外字書体として管理していた。

【0004】

【発明が解決しようとする課題】しかしながら、上記従来の文字処理装置における外字の管理、利用方式においては以下に示すような問題がある。

【0005】①文字として扱うべき外字には、明朝体、ゴシック体、毛筆体と言った書体の概念が存在する。このため、同じ文字パターンでも、そのデザインの種類が複数存在する場合がある。そして従来の文字処理装置においては、例えば明朝体の「う゛」、ゴシック体の「う゛」、毛筆体の「う゛」に対しては、同じ文字であるにも関わらず、別々の文字コードを割り付ける必要がある。このため、実際に使用する際に不便であるという問題がある。

【0006】②例えばワードプロセッサやコンピュータ上に動作するワープロソフトやDTPソフト等を用いて文章を作成する場合、文章の特定の範囲を選択して、その範囲の書体を変更するという操作がある。この様な操作の指定範囲内に外字が存在する場合、外字として作成した文字が書体毎に異なる文字コードに割付けられているため、外字の部分だけが意図した文字とは異なる文字になってしまったり、或いは空白になってしまう。

【0007】本発明は上記の問題点に鑑みてなされたものであり、書体の概念が存在する外字について、異なる書体の間の同一の外字に対して同種のコードを割り付けて管理することを可能とし、書体間における外字のコードの整合性を保つと共に、操作性を向上する文字処理方法及び装置を提供することを目的とする。

【0008】

【課題を解決するための手段】上記の目的を解決するための本発明による文字処理装置は以下の構成を備えている。即ち、ユーザがデザインした外字の登録及び利用が可能な文字処理装置であって、登録された外字と該外字が所属する書体とを対応づける対応情報を登録する登録手段と、前記所属する書体を異にする同種の外字に対して同一の識別コードを付与するコード付与手段と、前記所属する書体及び前記識別コードを用いて所望の外字を取得する取得手段と、を備える。

【0009】又、上記の目的を解決するための本発明の文字処理方法は以下の工程を備える。即ち、ユーザがデ

ザインした外字の登録及び利用が可能な文字処理方法であって、登録された外字と該外字が所属する書体とを対応づける対応情報を登録する登録工程と、前記所属する書体を異にする同種の外字に対して同一の識別コードを付与するコード付与工程と、前記所属する書体及び前記識別コードを用いて所望の外字を取得する取得工程と、を備える。

【0010】

【作用】上記の構成により、ユーザにより作成された外字は、その外字の属する書体とを対応づける対応情報が登録され、更に、書体の異なる同種の外字に対しては同一の識別コードが付与される。そして、書体及び識別コードを用いて取得すべき外字を指定することにより、所望の外字を獲得する。

【0011】

【実施例】以下に添付の図面を参照して、本発明の好適な実施例について説明する。

【0012】＜実施例1＞図1は本実施例における文字処理装置の基本的な構成を示すブロック図である。尚、本装置は、例えば、ワードプロセッサであっても良いし、パーソナルコンピュータ、ワークステーション或はコンピュータシステムであっても良い。

【0013】図1において、1は中央処理装置（CPU）であり、本装置全体の制御及び演算処理等を行なう。2は読み出し専用メモリ（ROM）であり、システム起動プログラム及び文字パターンデータ、文字コード情報等の記憶領域である。通常、ROM2に格納されている文字パターンは、システムフォントとしてメニューの文字表示等に利用される。文字コード情報はシステムフォントのコード体系の情報であり、新JIS、旧JIS等の情報を有する場合もある。尚、システムフォントはROM2上の書体データでも、予めユーザが指定した書体であってもかまわない。3はランダムアクセスメモリ（RAM）であり、使用制限のないデータ記憶領域である。即ち、RAM3には様々な処理毎に各々のプログラム及びデータがロードされる。

【0014】4はキーボード制御部（KBC）であり、キーボード5より入力データを受け取り、CPU1へ伝達する。6はディスプレイ制御部（CRTC）であり、CRT7に対して表示データを出力する。7はディスプレイ装置（CRT）であり、CRTC6よりデータを受け取りこれを表示する。8はディスク制御部（DKC）であり、外部記憶装置とデータの伝送を行う。9は外部記憶装置であり、フロッピーディスク装置（FD）或いはハードディスク装置（HD）等を備える。外部記憶装置9は、プログラム及びデータを格納しており、これらはCPU1による各種処理の実行時に、必要に応じて参照されるか、又はRAM3へロードされる。例えば、基本システムが必要な処理を要求すると、この外部記憶装置9からその処理を行なうシステム又はプログラムがR

AM3上にロードされ、CPU1により起動される。CPU1はシステム又はプログラムを実行する間に、必要に応じて外部記録装置9に格納されたデータや情報をRAM3上にロードする。又、後述のフローチャートで表される制御プログラムもRAM3上にロードされ、CPU1により実行される。

【0015】10はプリンタ制御部(PRTC)であり、プリンタ11に対して記録データを出力する。11はプリンタであり、PRTC10より受信した記録データに基づいて記録媒体上に可視画像を形成する。12はシステムバスであり、上述の構成要素間のデータの通路となるべきものである。

【0016】図2は、RAM3のデータ構成を表す図である。201はウインドウシステム領域であり、ウインドウシステムプログラムがロードされる領域である。202はウインドウシステムワーク領域であり、ウインドウシステムが種々の処理の実行時において使用するワーク領域である。203は管理領域であり、本ウインドウシステムで用いるプリンタドライバやCRTドライバ等の各種デバイスドライバに関する管理情報や、フォントドライバ等に関する管理情報を格納する。204はフォント管理情報テーブル領域であり、各フォントドライバが扱う書体やシステムフォント等、フォントに関連する情報を登録するフォント管理情報テーブルを格納する領域である。205はウインドウシステムキャッシュ領域であり、ウインドウシステムが管理する文字データのキャッシュ領域である。206はフォントドライバ領域であり、ウインドウシステムが用いるフォントドライバを格納する。207は相対テーブル領域であり、後述する相対テーブルを格納する領域である。

【0017】図3はウインドウシステムの基本的な構成を示すブロック図である。301はウインドウシステム(OSを含んでもよい)であり、RAM3のウインドウシステム領域201上にロードされ、CPU1によって起動される。302はフォント管理情報テーブルであり、ウインドウシステム301が統括するフォントドライバの有する書体に関する情報や、アプリケーションより取得要求のあった文字に関する情報等を有し、RAM3のフォント管理情報テーブル領域204に格納される。又、303及び304はウインドウシステム301より起動されるアプリケーションであり、それぞれアプリケーションA、アプリケーションBとする。

【0018】各種デバイスドライバ305~309は本システムを構成する各デバイスを制御するプログラムである。即ち、プリンタドライバA305、プリンタドライバB306はプリンタを制御するプログラムである。又、CRTドライバ307はCRTを、キーボードドライバ308はキーボードを、マウスドライバ309はマウスをそれぞれ制御するプログラムである。これら各種デバイスドライバ305~309はRAM3上にロード

されて、管理領域203に格納された管理情報を基にウインドウシステム301に管理されている。

【0019】また310及び311はそれぞれフォントドライバA及びBであり、要求された文字を作成して提供する為の手段であり、装置として提供されても、プログラムにて提供されてもよい。プログラムにて提供された場合は、例えばRAM3上にロードされた後実行される。また、フォントドライバ310、311は直接ウインドウシステム301に管理されても、プリンタドライバ305、306やディスプレイドライバ307等の各種デバイスドライバに管理されても、包含されても良い。

【0020】図4及び図5は、ウインドウシステムの起動手順、及びアプリケーションからの文字取得要求に対する処理手順を示すフローチャートである。

【0021】ステップS11において、ウインドウシステム301がRAM3のウインドウシステム領域201にロードされ、CPU1によりウインドウシステムが起動される。このとき、RAM3上にウインドウシステムワーク領域202が確保されると共に、ウインドウシステムが統括すべき各種デバイスドライバ及びフォントドライバの数量や状態に基づいて管理領域203が確保される。更に、フォント管理情報テーブル領域204とウインドウキャッシュ領域205がRAM3上に確保され、その内容が初期化される。

【0022】次に、ステップS12において、各フォントドライバ(310~311)がRAM3のフォントドライバ領域206上にロードされるとともに起動される。そして、ステップS13において、ウインドウシステムの初期画面をCRT7に表示する。ステップS14において、各フォントドライバ(310~311)に対して書体情報の要求が行われる。ステップS15においては、ステップS14で各フォントドライバ(310~311)より提供された情報を基に書体の情報テーブル(フォント管理情報テーブル302の一部)を作成し、RAM3上のフォント管理情報テーブル領域204に格納する。以上のようにしてウインドウシステムは初期画面の表示を含む初期動作を完了する。そして、ステップS16において、キーボードやマウス等の入力デバイス5からの各種イベント待ち、即ち、アプリケーションの起動イベント待ち状態となる。

【0023】アプリケーション(303、304等、一つ或いは複数)が起動されるとステップS17へ進み、文字の取得要求や書体、サイズの設定要求等のイベント待ち状態となる。そして、アプリケーションのイベントが発生するとステップS18へ進む。ステップS18以降の処理は、アプリケーションのイベント発生に対する処理であり、ここでは、ウインドウシステムが管理する複数のアプリケーションのうちの1つのアプリケーション(ここではアプリケーションA303)の処理動作に

限定して説明する。尚、複数のアプリケーションが起動している状態では、ステップS18以降の処理が、各アプリケーション毎に独立して存在し、ウインドウシステムはそれぞれを管理している。

【0024】ステップS18では、アプリケーションA303内の処理として発生したイベントが、フォントドライバ(310、311等一つ或いは複数)に対する書体情報の要求であるか否かを判断する。ここで、書体情報の要求である場合は、制御はステップS22へ移行する。ステップS22において、ウインドウシステム301は、本ウインドウシステム301が管理する全てのフォントドライバ(310、311)に対して、フォントドライバが所有する全書体の情報要求を発行する。各フォントドライバは、ウインドウシステムからの情報要求に応え、所有する全書体の情報(書体情報)をウインドウシステムに転送する。各フォントドライバからの書体情報を受け取ったウインドウシステムは、その全ての書体情報を、アプリケーションA303に転送する。これら一連の処理が、ステップS22における、書体情報をアプリケーションに転送する処理であり、ステップS22の処理を実行後、アプリケーションのイベント待ちであるステップS17に戻る。

【0025】ステップS18において、発生したイベントが書体情報の要求以外であったときは、次の処理であるステップS19へ進む。

【0026】ステップS19は、発生したイベントが文字の取得要求(ウインドウシステムに対する文字要求)であるか否かを判断する。文字の取得要求でない場合はステップS23に移行し、発生したイベントに対応した各種の処理を実行し、アプリケーションのイベント待ちであるステップS17に戻る。又、ステップS19にて文字の取得要求(ウインドウシステムに対する文字提供)のイベントが発生していた場合は、次の処理であるステップS20に進み、文字の取得処理を実行する。

【0027】ステップS20において、アプリケーションA303が要求した文字データ(サイズ指定、書体選択を含む、諸情報の要求)が、ウインドウシステム301が管理するウインドウシステムキャッシュ領域205内に存在するか否かを判断する。これはフォント管理情報テーブル302を検索することで実施する。そして、文字データがキャッシュ領域内に存在する場合はステップS21に進み、ウインドウシステム301はウインドウシステムキャッシュ領域205内に存在する文字をCRTドライバ307等に渡し、出力要求を行う。一方、指定された文字データがウインドウシステムキャッシュ領域205内に存在しない場合は、ステップS20からステップS24に移行する。

【0028】ステップS24において、指定された書体の名称、サイズから、どのフォントドライバに対して文字の出力要求を行うべきであるかを決定する。そして、

ステップS25において、アプリケーションから要求された「書体」及び「サイズ」を単位として、文字の出力要求を行ったフォントドライバからの情報をフォント情報管理テーブル302に登録する。フォント管理情報テーブル302は、「書体」と「サイズ」を単位として、ウインドウシステムが管理する全てのアプリケーションから要求があった文字について管理する。ステップS26では、ステップS24にて決定したフォントドライバに対して文字の出力要求を行う。ここで、出力要求を受けたフォントドライバは、要求に従った文字をウインドウシステム301に提供する。ステップS27では、ステップS26にてフォントドライバより提供された文字パターンを、ウインドウシステム301が管理するウインドウシステムキャッシュ領域205内に格納する。

【0029】その後、ステップS21に進み、ディスプレイ等に文字を出力し、アプリケーションのイベント待ちのステップS17に戻る。このステップS17からステップS21までの処理が、アプリケーションが終了するまでの間継続される。

【0030】次に、ウインドウシステム301より起動されたアプリケーションの文字取得に関わる基本的な動作について、図6のフローチャートを用いて説明する。尚、ここでは、1例として、アプリケーションA303が起動されたものとして説明する。

【0031】ウインドウシステム301より、アプリケーションA303の起動要求イベントが発行されると、ステップS31において、初期画面の表示が行われる。この初期画面により、アプリケーションA303が使用可能状態にあることがユーザに告知される。その後ステップS32において、アプリケーションA303は、現状にて使用可能な書体に関する全ての情報を取得するため、ウインドウシステムが管理する全ての出力デバイスのドライバ(プリンタドライバ305、306、ディスプレイドライバ307等)に対し、ウインドウシステム301を介して問い合わせを行う。そして、アプリケーションA303は得られた使用可能な書体名を一覧表示した書体リスト(図7)を作成し、表示する。

【0032】ステップS33では、ユーザに書体リスト(図7)の中から所望の書体を選択させるとともに、文字に対するあらゆるパラメータを設定させる。ここで文字に対するパラメータとは、文字のサイズや文字の間隔(ピッチ)、行間角(ラインピッチ、行ピッチ)、修飾や文字の太さ等の情報をユーザに指定させる。

【0033】ステップS34において、キーボード5等から文字(ひらがな、カタカナ、漢字、英数字や記号等)が入力されると、ステップS35へ進む。ステップS35では、ステップS33で設定された書体及びパラメータに従って、ステップS34にて入力された文字を取得するため、ウインドウシステム301に文字の出力要求を行う。いわゆる文字取得(出力)処理とは、ステ

ップS34からステップS35であり、アプリケーションが実行されている間、通常この処理が繰り返し実行される。

【0034】次に、本実施例1が行う文字の管理方法の概要について説明する。本実施例1では、ユーザが独自に作成したパターン（以降、外字と称する）を格納する外字ファイル（図8）が、複数の通常の書体ファイル（図9）と合わせて存在する場合を例に挙げて説明する。本実施例1においては、外字ファイル（図8）内の各文字を、通常の書体ファイル（図9）に関連付けるための相対テーブル（図10）が用いられる。即ち、本実施例1における文字及び書体の管理方法とは、相対テーブルを用いて、外字ファイル21内の各外字を通常の書体に従属させて管理する手法である。

【0035】図8は、ユーザが作成した文字（外字）をを格納する外字ファイルの内容を説明する図である。21は外字ファイル全体を示している。又、外字ファイル21の上段に付された番号は、文字コードとは無縁の外字管理番号21aであり、この外字ファイル21内でのみユニークな番号である。そして、各外字管理番号21aに対応して外字パターン21bが1つずつ登録される。

【0036】図9は、通常の書体ファイルのデータ構造を示す図であり、文字コード31aに対応して各文字データ31bが格納されている。書体ファイルとしては、明朝体ファイル31、ゴシック体ファイル32、毛筆体ファイル33がある。

【0037】図10は、外字ファイル内の各外字と通常の書体ファイルとを関連づけるための相対テーブルのデータ構造を表す図である。相対テーブル41は、外字ファイル21に登録されている各外字をいかなる書体に従属させ、かつ、如何なる文字コードに対応させるかを管理する為のテーブルである。図10において、41aは相対テーブルにおける順番を表すシーケンス番号である。41dは外字ファイル内番号であり、外字ファイル21における外字管理番号21aと対応している。41bは書体名称であり、外字ファイル内番号41dが示す外字を従属させる書体を識別するための情報である。41cは文字コードであり、外字ファイル内番号41dが示す外字に割り当てられた文字コードを登録する。

【0038】例えば、図8の外字ファイル21において、外字管理番号21aが1番の「う゛」は明朝体、5番の「う゛」はゴシック体、16番の「う゛」は毛筆体であるとし、文字コードをJISコードの「2147」に割り付けたとする。この場合、明朝体の「う゛」についての情報が、相対テーブル41の1番の欄の如く登録される。同様に、外字ファイル21の5番のゴシック体の「う゛」は相対テーブル41の5番の欄、外字ファイル21の16番の毛筆体の「う゛」は相対テーブル41の16番の欄のように、それぞれ所定の情報が登録され

る。

【0039】本実施例1の文字処理装置では、例えばゴシック体の文字コード「2147」の文字に対する取得要求が発行された場合、相対テーブル41に該当する書体の文字が有るか否かを検索する。即ち、書体名称41bが「ゴシック対」であり、文字コード41cが「2147」である外字を検索する。そして、相対テーブル内に該当する外字が存在した場合には外字ファイル21より文字情報を取りだして文字パターンに展開する。このようにして、ゴシック体による「う゛」という文字が取得される。

【0040】又、外字ファイル21中の6番から15番のような、書体としての概念を伴わず、記号や図形として扱うべき外字については、「外字書体」のように単独の独立した書体として定義し、ユーザに識別可能な別書体として管理する。

【0041】図11は、図8に示す外部ファイル21と、図9に示す通常の書体ファイル31、図10に示す相対テーブル41との関連を簡単に表した概念図である。こうした関連性を付けた管理を実現することが、本実施例1の文字処理装置の根幹を成す概念であり、処理、管理方法そのものである。

【0042】次に、上述の構成を備える本実施例1の文字処理装置における、外字の登録及び取得処理について更に説明する。

【0043】まず、図12のフローチャートを基に、作成した外字の登録処理について説明する。図12は、実施例1の文字処理装置における登録処理の処理手順を表すフローチャートである。本フローチャートは、ユーザが外字を作成した後に、当該文字処理装置に対して登録要求を発行することにより実行される。尚、登録要求において、登録すべき外字とその書体及び文字コードが指定される。

【0044】まず、ステップS41においては、既に1つ以上の外字が登録されているか否かを判断し、登録された外字が存在しない場合はステップS42へ進む。ステップS42では、外字を管理するための相対テーブル41を作成するための領域を、RAM3上に確保すると共に初期状態にセットし、ステップS45へ進む。

【0045】一方、ステップS41において何らかの外字が既に登録されている場合はステップS43へ進む。ステップS43では、指定された書体及び文字コードにより相対テーブル41を検索する。そして、ステップS44では、先のステップS43にて検索した結果を基に、該当する外字が既に存在しているか否かを判断し、存在しない場合はステップS45へ、存在している場合はステップS47へそれぞれ進む。

【0046】ステップS45では、作成した外字の新規登録を行なう。ここでは、登録すべく指定された外字を、外字ファイル21の末尾に追加する。尚、ここで

は、作成された外字を、外字ファイル21の末尾に追加しているがこれに限られるものではない。例えば外字ファイル21の先頭に挿入したり、出現頻度や文字コードの昇順降順等に基づく適切な位置に挿入するようにしたりしてもよい。ステップS46では、ステップS45で登録した外字の書体及び文字コード等の情報を相対テーブル41に登録し、本処理を終了する。

【0047】一方、ステップS44において、該当する外字が存在する場合はステップS47へ進む。ステップS47以降の処理は、以前に作成、登録された外字の更新処理にあたる。ステップS47では、外字ファイル21内の該当する外字を一旦削除し、新たに作成された外字を登録する。そして、ステップS48では、相対テーブル41上の該当する箇所を、新たに登録された外字に関する情報（書体、文字コード等）で更新する。即ち、相対テーブル内の該当箇所に存在していた外字に関する相対情報を一旦削除し、今回新たに登録した外字に関する相対情報を登録する。

【0048】尚、図12のフローチャートには記載されていないが、作成した外字を削除する場合は、指定された外字について、外字ファイル21より文字を削除すると共に相対テーブル41からも対応する相対情報を削除する処理を、同様な手続きで行なうことが可能であることは明かである。

【0049】こうした処理を繰り返す事で、外字の新規作成、更新／変更／修正、削除に対応し、常に最新の状況を把握、管理する事が可能となる。

【0050】次に、図13のフローチャートを参照して、文字イメージを取得する際の処理手順について説明する。図13は、本実施例1の文字処理装置における文字パターンを生成するための処理手順を表すフローチャートである。尚、本処理は、文字パターンの取得要求が発行された時点から実行される。

【0051】ステップS51において、指定された書体及び文字コードにて相対テーブル41を検索する。ステップS52では、ステップS51における検索結果により、指定された文字が外字であるか否かを判断する。即ち、指定された文字が相対テーブル41内に存在すれば外字であると判断し、存在しなければ通常の文字であると判断する。ステップS52における判断の結果、外字であればステップS54へ、外字でなければステップS53へそれぞれ進む。

【0052】ステップS53では、通常書体ファイル31より文字データを取得し、文字パターンを生成し出力する。

【0053】ステップS54では、ステップS51における相対テーブル41の検索結果より、指定された書体及び文字コードを有する外字のパターンデータの外字ファイル21内の格納場所を特定する。そして、ステップS55にて、外字ファイル21より該当する外字のパター

ンデータを取得し、文字パターンを生成し、出力する。

【0054】以上のようにして、外字ファイル21、通常書体ファイル31、相対テーブル41の各データを用いることにより、通常書体に従属させる外字と、外字のみの集合体である外字書体を区別して管理することが可能となる。

【0055】以上説明したように、実施例1によれば、文字として扱うべき外字は書体別に通常書体に従属させ、簡単なイラストや記号等の文字として扱わない外字のみを独立した単独の書体として扱うことが可能となる。そして、文字として扱う外字については書体間で同一の文字コードが付与されるので、書体間の外字の整合性を保つことが可能となる。又、取得すべき文字パターンとその文字コードとの関連をユーザの記憶に頼ることなく、文字処理装置内で管理することが可能となるので操作性が向上する。

【0056】＜実施例2＞上記実施例1では、外字ファイル内に格納されている各外字を、相対テーブルにより、通常書体（明朝体、ゴシック対等）と、外字書体（イメージパターンとして扱う外字）とに区別して管理している。実施例1の文字処理装置においては、文字パターンの取得及び登録時に相対テーブル41を検索するため、外字の登録数が多くなると検索時間がかかり、処理速度の低下が起きる。

【0057】本実施例2では、実施例1と同様に外字を各書体に従属させて管理すると共に、目的とする文字の取得を高速に行い処理のスループットを向上する文字処理装置について説明する。尚、実施例2の文字処理装置の構成及びウインドウシステムの基本的な動作については実施例1と同様であるのでここではその説明を省略する。

【0058】図14は実施例2における文字処理装置のRAM3のデータ構成を表す図である。実施例1と比べて、インデックステーブル領域208が追加されている。インデックステーブル領域208にはインデックステーブルが格納される。

【0059】インデックステーブルとは、文字データの格納場所をサーチする際の検索処理を高速に行うために用いられるテーブルである。このインデックステーブルは、相対テーブル内に格納されている各種の情報について、その各種項目をキーに検索するための、独立した個別のテーブル群である。例えば書体名をキーに検索する場合は、図15の（A）～（C）に示す書体別インデックステーブルを用いる。又、文字コードをキーに検索する場合は図16に示す文字コード別インデックステーブルを用いる。このようにインデックステーブルを検索することにより、相対テーブルのどこに必要とする文字データの情報が格納されているかを取得する。その後、相対テーブルの情報を読み出して、外字ファイルの該当する文字データを取得して文字パターンを生成する。

【0060】尚、図15の(A)は明朝体に割り付ける外字の情報のみを集めたインデックステーブル51であり、(B)はゴシック体に割り付ける外字の情報のみを集めたインデックステーブル52であり、(C)は外字書体に割り付ける外字の情報のみを集めたインデックステーブル53である。各インデックステーブルにおける相対テーブル内番号とは、相対テーブル41のシーケンス番号41aに対応しており、相対テーブル41内の参照すべき位置を示すものである。

【0061】尚、図15は、各書体別に独立したインデックステーブルを作成した場合を示すが、これらをまとめて一体とした形式でも実現可能である。本実施例2では、外字の文字データを格納した外字ファイルが1つであるケースを想定したものであるが、複数の外字ファイル21が存在してもよい。この場合、相対テーブル41に登録する項目として外字ファイルを識別するための番号を追加することで実現可能である。

【0062】次に実施例2の文字処理装置の動作について説明する。

【0063】まず、図17のフローチャートを参照し、作成した外字の登録処理について説明する。図17は、実施例2の文字処理装置における登録処理の処理手順を表すフローチャートである。本フローチャートは、ユーザが外字を作成した後に、当該文字処理装置に対して登録要求を発行することにより実行される。尚、登録要求において、登録すべき外字とその書体及び文字コードが指定される。

【0064】まず、ステップS61においては、既に1つ以上の外字が登録されているか否かを判断し、登録された外字が存在しない場合はステップS62へ進む。ステップS62では、外字を管理するための相対テーブル41及びインデックステーブルを作成するための領域を、RAM3上に確保すると共に初期状態にセットし、ステップS65へ進む。

【0065】一方、ステップS61において何らかの外字が既に登録されている場合はステップS63へ進む。ステップS63では、指定された書体及び文字コードによりインデックステーブルを検索し、指定された書体及び文字コードで既に登録された外字が存在するか否かを調べる。ここで、使用するインデックステーブルは書体別インデックステーブルでも文字コード別インデックステーブルでもよい。ステップS64では、先のステップS63にて検索した結果を基に、該当する外字が既に存在しているかを判断し、存在しない場合はステップS65へ、存在している場合はステップS67へそれぞれ進む。

【0066】ステップS65では、作成した外字の新規登録を行なうために、登録すべく指定された外字を、外字ファイル21の末尾に追加する。尚、ここでは、作成された外字を、外字ファイル21の末尾に追加している

がこれに限られるものではない。例えば外字ファイル21の先頭に挿入したり、出現頻度や文字コードの昇順、降順等に基づく適切な位置に挿入もしくは並べ替え(ソート処理)するようにしてもよい。ステップS66では、ステップS65で登録した外字の書体及び文字コード等の情報を相対テーブル41に登録すると共に、インデックステーブルにも登録し、本処理を終了する。尚、インデックステーブルは、検索効率の向上を目的とした情報テーブルであるため、新たに外字が追加された場合には、キーとなる「書体名称」や「文字コード」等による昇順、降順等の規則に従ったデータの並び替え(ソート処理)を行ってもよい。

【0067】一方、ステップS64において、該当する外字が存在する場合はステップS67へ進む。ステップS67以降の処理は、以前に作成、登録された外字の更新処理にあたる。ステップS67では、外字ファイル21内の該当する外字を一旦削除し、新たに作成された外字を登録する。そして、ステップS68では、相対テーブル41及びインデックステーブル上の該当する箇所を、新たに登録された外字に関する情報(書体、文字コード等)で更新する。即ち、相対テーブル及びインデックステーブル内の該当箇所に存在していた外字に関する情報を新たに登録した外字に関する情報に更新する。又、ここで、インデックステーブルに対してソート処理等を行ってもよいことは言うまでもない。

【0068】次に、図18のフローチャートを参照して、文字イメージを取得する際の処理手順について説明する。図18は、本実施例2の文字処理装置における文字パターンを取得するための処理手順を表すフローチャートである。尚、本処理は、文字パターンの取得要求が発行された時点から実行される。

【0069】ステップS71において、指定された書体及び文字コードにて文字コード別インデックステーブルを検索する。ステップS72では、ステップS71における検索結果により、指定された文字が外字であるか否かを判断する。即ち、指定された文字が文字コード別インデックステーブル内に存在すれば外字であると判断し、存在しなければ通常の文字であると判断する。ステップS72における判断の結果、外字であればステップS74へ、外字でなければステップS73へそれぞれ進む。

【0070】ステップS73では、通常書体ファイル31より文字データを取得し、文字パターンを生成し出力する。

【0071】ステップS74では、ステップS71における検索結果を基に、相対テーブル41内の指定された外字に関する情報の格納場所を特定する。そしてステップS75にて、相対テーブル41より目的とする外字の情報を取得し、パターンデータが、外字ファイル21内の何処に格納されているかを特定する。そして、ステッ

ブS76にて、該当する外字のパターンデータを取得し、文字パターンを生成し、出力する。

【0072】以上説明したように実施例2の文字処理装置によれば、文字コードと文字パターンの整合性を実現することが可能となると共に、所得したい文字パターンとその文字コードとの関連を操作者の記憶に頼ることなく、文字処理装置内にて効率よく管理することが可能となる。更に、管理した各外字の文字パターン展開処理を、インデックステーブルを作成、活用することで、より高速に実行することが可能となる。

【0073】尚、備えるべきインデックステーブルは書体別インデックステーブル、文字コード別インデックステーブルのどちらか一方のみでもよい。

【0074】<実施例3>上記実施例1及び2においては、例えば明朝体のデザインによる「うゝ」のみを作成して、明朝体に従属登録した状態で、ゴシック体の「うゝ」を用いようとする、ゴシック体の「うゝ」が未作成であるため、文字の出力ができない。即ち、外字として作成した「うゝ」を含む明朝体で入力した文章を、そっくりゴシック体に変更した場合、外字の「うゝ」のみが抜けてしまう。

【0075】従って、本実施例3では、同じ文字コードに割り振られた外字が他の書体に存在する場合、未作成の書体の外字を現存する他の書体の文字で代用することで、書体間の整合を保ち、書体を変更しても、未作成文字の空白表示や異なる文字が出力されることを防止する。

【0076】本実施例3においても上述の実施例2と同様に、外字ファイル21内の各外字を、相対テーブルを用いて通常の書体に従属させて管理する。但し、本実施例3の相対テーブルには代用情報が含まれる。図19は実施例3の相対テーブルのデータ構成を表す図である。同図において、相対テーブル内の2、3、4、20番目に示す如く（代用）と記述された欄は、相当する書体用の外字が未作成であるため、既存の外字で代用していることを示す代用情報である。このように、ユーザが作成した外字を書体に従属させる際に、作成された外字と、他の書体の対応する（同一文字コードに該当する）未作成の文字を対応させる。

【0077】本実施例3では書体の概念が存在する、文字として扱う外字について、使用可能な全書体（本例では、明朝体、ゴシック体、毛筆）について同一文字コードを有する外字が作成されていることが理想的であるとして、未作成の書体の外字について代用する文字を割り付ける機能を有する。

【0078】図20は実施例3における相対テーブルと各書体間の関連を表す図である。

【0079】更に、実施例2と同様に、上位より発行された文字パターンの取得要求に応じて文字パターンを展開する処理において、その処理速度を向上させるために

各種のインデックステーブルを設ける。実施例2と同様に書体名別インデックステーブル、文字コード別インデックステーブルを備える。図21は図19に示された相対テーブルについてのインデックステーブルであり、

（A）は明朝体に割り付ける外字の情報のみを集めたインデックステーブルであり、（B）はゴシック体に割り付ける外字の情報のみを集めたインデックステーブルであり、（C）は外字書体に割り付ける外字の情報のみを集めたインデックステーブルである。（B）のゴシック体用のインデックステーブルにおいて、2、7、8番がゴシック体として作成された外字が存在しないため、明朝体として作成された外字で代用するための情報を格納した状態を表す。

【0080】又、図22は実施例3における文字コード別インデックステーブルである。（代用）が記されている欄が代用情報を表している。図23は、代用文字別インデックステーブルのデータ構成を表す図である。代用の対象となる文字をキーに検索する場合は代用文字別インデックステーブルを用いる。

【0081】尚、本例では各書体別に独立したインデックステーブルを作成した場合を例としているが、それらをまとめて一体とした形式でも実現可能である。更に、本実施例では、外字の文字データを格納した外字ファイルが1つであるケースを想定しているが複数のファイルが存在した場合においても、相対テーブル内に格納する項目に文字ファイル番号を追加する等、各インデックステーブルにも相当する情報項目を追加することで実現可能である。

【0082】次に実施例3における文字処理装置の動作について説明する。図24及び図25は実施例3の文字処理装置における外字登録の手順を表すフローチャートである。

【0083】まず、ステップS101においては、既に1つ以上の外字が登録されているか否かを判断し、登録された外字が存在しない場合はステップS102へ進む。ステップS102では、外字を管理するための相対テーブル及び各種インデックステーブルを作成するための領域を、RAM3上に確保すると共に初期状態にセットしステップS111へ進む。

【0084】まず最初に、登録したい書体の書体別インデックステーブルを指定された文字コードで検索する。そして、ステップS104において、ステップS103の検索の結果から指定された文字コードに該当する外字が既に存在するか否かを判断し、存在する場合はステップS105へ、存在しない場合はステップS111へそれぞれ進む。

【0085】ステップS111では、外字ファイルへ新たな外字を追加登録し、ステップS112では、相対テーブルへ新たな外字に関する情報を追加登録する。更に、ステップS113で、指定された書体に対応する書

体別インデックステーブルへ新たな外字に関する情報を追加登録する。又、ステップS114では、文字コード別インデックステーブルへ新たな外字に関する情報を追加登録する。ステップS115では、新規に登録された新たな外字を他の書体で代用する可否かを判断する。ここで、代用する可否かは、ユーザの指定により判断しても、自動的に判断してもよい。又、無条件に代用を実行するようにしてもよい。

【0086】ステップS115において、代用すると判断された場合は、ステップS116へ進み、相対テーブル及び各種インデックステーブルに代用情報を追加登録する。即ち、相対テーブルに代用する書体の名称及び外字ファイル内の番号等、代用情報として必要な情報を付加する。続いて、各インデックステーブルについて必要な情報を追加する。

【0087】例えば、各書体別インデックステーブルについて、オリジナルの書体別インデックステーブルに作成した外字に関する情報を追加する。例えば、ゴシック体用にデザインした外字を登録する場合は、オリジナルの書体別インデックステーブルはゴシック体の書体別インデックステーブルとなる。そして、作成した外字を他の書体で代用して使用する場合には、各書体（明朝体、毛筆体）の書体別インデックステーブルに、今回作成した外字を代用して使うための情報を追加登録する。同様にして、文字コード別インデックステーブルや代用別インデックステーブルにも必要な情報を追加する。

【0088】一方、ステップS104で、重複する外字があると判断された場合は、ステップS105へ進む。ステップS105では、検索された外字がオリジナルであるか代用であるかを判断し、オリジナルであればステップS108へ、そうでなければステップS106へそれぞれ進む。

【0089】ステップS108では、現存する外字を破棄し、新たに作成した外字に更新する否かを判断する。この更新の可否は、変更してよいかどうかをユーザに指定させる操作を促してユーザに判断させても、自動的に判断してもよい。又、指定された書体の指定された文字コードに既に作成された外字が登録されていることを告知し、新たな文字コードに（外字が存在していない領域）変更して、登録する可否かをユーザに判断させる処理を付加してもよい。

【0090】ステップS108で更新すると判断された場合はステップS109へ進み、更新しないと判断された場合は本処理を終了する。ステップS109では、書体別インデックステーブルを検索した結果に基づいて相対テーブルを参照し、外字ファイル内の該当する外字を削除し、新たな外字を格納する。続いてステップS110において、相対テーブルの内容及び各種インデックステーブルの対応する部分について、必要な情報の修正を行う。

【0091】次に、ステップS105において、重複する外字が代用の外字であった場合について説明する。この場合処理はステップS106へ進む。

【0092】この場合、重複する外字は代用の外字であるため、外字ファイル内には、新たな外字にともなって削除される外字は存在しない。従って、まずステップS106にて外字ファイルの末尾に、今回作成した外字を格納する。そして、ステップS107へ進み各種テーブルの更新を実行する。例えば、相対テーブルについては、代用情報を抹消して、今回追加した外字に関する情報を登録する。各種インデックステーブルについても同様に、代用の情報を抹消し、新たに作成した文字を使用するように更新する。

【0093】こうした処理を繰り返すことで、外字の新規作成、更新／変更／修正、削除に関して、常に最新の状況を把握、管理することが可能となる。

【0094】尚、外字の取得手順に付いては上述の実施例2と同様であり、ここでは説明を省略する。

【0095】＜実施例4＞上記各実施例では、文字として扱うべき外字と、イメージとして扱うべき外字とを、区別なく一律に展開してパターン情報を得ている。

【0096】特に、文字サイズを自由に設定し、いかなる大きさの文字パターンを取得することが可能なスケラブルフォントの場合は、取得したいサイズによって文字がつぶれたように変形してしまうため、文字パターンを構成する部首等には線分等の間隔や太さを均一にし、バランスや可読性を保つための修正情報（ヒント情報）が文字データに付加される場合が多い。この情報は、展開するパターンが文字であることを前提とした情報であるため、書体の概念が伴う文字として扱う外字には適用させることが可能であるが、イラストなどのイメージとして扱う適用が不可能な情報である。

【0097】従って外字として作成した文字には、ヒント情報を一切付加せずに文字データを作成するケースが最も多く、可読性や文字のバランスが悪いパターンを展開している。

【0098】又、絵文字のように、図形や記号、干支やイラスト等、イメージとして扱う文字について通常の書体の概念が伴う文字と同様なヒント情報に類する修正、補正情報の付加は極めて困難である。その理由として、イメージ全体のバランスを損ない、オリジナルのイメージを忠実に再現できないケースがあるためである。

【0099】本実施例4では、文字として扱うべき外字と、イメージとして扱うべき外字とを区別して管理し、文字として扱うべき外字については通常の文字と同様なヒント情報を用いて、バランスのよい可読性の高い高品位の文字パターンを展開する。又、イメージとして扱うべき外字については、オリジナルのイメージに忠実なパターンを展開するため、ヒント情報を用いずにパターンへの展開を行う。尚、イメージデータの展開において

は、通常の文字と異なる絵文字用のヒント情報を用いるようにしてもよい。

【0100】次に、実施例4の文字処理装置における外字管理の手順に付いて説明する。尚、外字の登録手順は上記実施例2と同様であり、ここでは説明を省略する。従って、以下に外字を取得するための手順について説明する。図26は実施例4の文字処理装置の外字取得手順を表すフローチャートである。

【0101】ステップS121において、指定された書体及び文字コードにより文字コード別インデックステーブルを検索する。ステップS122では、ステップS121における検索の結果を基に、指定された文字が、外字であるか通常の文字であるかを判断する。そして、外字であればステップS124へ、そうでなければステップS123へそれぞれ進む。ステップS123では、通常の書体ファイルより文字データを取得し、文字パターンを生成し出力する。

【0102】ステップS124では、文字コード別インデックステーブルの検索結果に基づいて相対テーブル41を検索し、指定された書体の指定された外字のデータが、外字ファイル内の何処に格納されているかを探す。そして、ステップS125で、該当する外字を外字ファイルより読み出し、展開用のワーク領域をRAM3上に確保する等、パターンへの展開に必要な準備を行う。尚、ワーク領域は、RAM3上の領域の一部を新規に確保してもよいし、予め確保された固定の領域を設けるようにしてもよい。

【0103】次にステップS126において、指定された外字が書体の概念を有する外字であるか、イメージとして扱うべき外字であるかを判断する。ここでは、指定された書体名が明朝体、ゴシック体、毛筆のいずれかであれば、文字として扱うべき外字であると判断し、外字書体であればイメージとして扱うべき外字であると判断する。文字として扱うべき外字であると判断された場合はステップS127へ進み、先のステップS125で読みだしたデータを通常の文字の展開時に使用するヒント情報を用いてパターンを生成する。一方、イメージとして扱うべき外字である場合はステップS128へ進み、先のステップS125で読みだしたデータに対して、ヒント情報は使用せずにパターンへの展開を行う。尚、ここで絵文字専用のヒント情報を用いるようにしてもよい。

【0104】以上説明したように本実施例4によれば、通常の書体に従属させる外字と、外字のみの集合体である外字書体を区別して管理することが可能となると共に、文字として扱うべき外字に関しては、ヒント情報を用いてパターン展開することが可能となる。このため、バランスのよい、可読性の高い、高品位の文字パターンに外字を展開することが可能となる。そして、イメージとして扱うべき外字（即ち外字書体に従属する外字）については、イメージ専用のヒント情報、もしくはヒント

情報無しでパターンへの展開を実行することで、オリジナルのイメージに忠実なパターンを展開することが可能となる。

【0105】上記各実施例において、外字ファイル21、書体ファイル31、相対テーブル41等は、RAM、ハードディスク、或いはROM等のいかなるメモリ上にあってもよい。又、ネットワークにて接続された他のシステム上に存在しても良い。さらに、対応する書体や、割り付ける文字コードの領域は任意の設定が可能であることは言うまでもなく、又、その設定については、ユーザが指定しても、システムが自動的に行なっても良い。

【0106】又、本発明はインテリジェント化されたデバイスも、システムが自動的に行なっても良い。又、本発明はインテリジェント化されたデバイス上（例えばプリンタやプロッタ）での実現も可能であることは言うまでもない。

【0107】この様に、本発明は、文字コードと文字パターンの整合性を実現することが可能となると共に、所望したい文字パターンとその文字コードとの関連を操作者の記憶に頼ることなく、文字処理装置内にて効率よく管理することが可能となる。

【0108】特に、パソコン、ワープロ、DTP装置等において、作成した文章の特定な範囲について書体を変更するケースが、頻繁に発生するシステムの場合、各書体における文字コードと出力される文字の整合性を実現させるためにも重要な技術となることは明かであり、こうした状況下において、本発明を使用した場合にこそ絶大な効果を発揮する。

【0109】更に、上述の様な効果を実現するための相対テーブルの構造や構成等について、他の手法や組合せでも実現が可能であるため、そのバリエーションについて以下に説明する。

【0110】上述の実施例では、作成した外字の文字データを格納する「外字ファイル」に対して、1つの相対テーブルを作成することで外字の管理を行なっているが、各書体毎に相対テーブルを作成しても、同様な管理を行なうことが可能である。

【0111】即ち、図3における「明朝体」に従属する外字群は、図2における外字ファイル内の1番目と2番目、3番目、4番目、20番目、21番目、22番目、23番目、…であり、図3における「ゴシック体」に従属する外字群は、図2における外字ファイル内の5番目と20番目、24番目、25番目、26場面、…書体の概念の存在しない記号集とでも言うべき「外字書体」として扱う外字群は図2における外字ファイル内の6番目と7番目、8番目、9番目、10番目、11番目、12番目、13番目、14番目、15番目、…と言った帰属する又は従属する書体別に相対テーブルを作成し管理する方法でも本発明を実現させることが可能である。

【0112】尚、本発明は、複数の機器から構成されるシステムに適用しても1つの機器からなる装置に適用しても良い。また、本発明はシステム或いは装置に本発明により規定される処理を実行させるプログラムを供給することによって達成される場合にも適用できることはいうまでもない。

【0113】

【発明の効果】以上説明したように、本発明の文字処理方法及び装置によれば、書体の概念が存在する外字について、異なる書体の間の同種の外字に対して同一のコードを割り付けて管理することが可能となり、書体間における外字のコードの整合性を保つと共に、操作性を向上することが可能である。

【0114】

【図面の簡単な説明】

【図1】本実施例における文字処理装置の基本的な構成を示すブロック図である。

【図2】実施例1の文字処理装置におけるRAM上のデータ構成を表す図である。

【図3】ウインドウシステムの基本的な構成を示すブロック図である。

【図4】ウインドウシステムの起動処理及び文字取得要求に対する処理手順を示すフローチャートである。

【図5】ウインドウシステムの起動処理及び文字取得要求に対する処理手順を示すフローチャートである。

【図6】アプリケーションの文字取得に関わる基本的な動作を表すフローチャートである。

【図7】アプリケーションの使用可能な書体を表す書体リストを示す図である。

【図8】外字ファイルのデータ構成を表す図である。

【図9】通常の手体ファイルのデータ構成を表す図である。

【図10】相対テーブルのデータ構成を表す図である。

【図11】外字ファイル、書体ファイル及び相対テーブルの関連を説明する図である。

【図12】実施例1の文字処理装置における登録処理の処理手順を表すフローチャートである。

【図13】本実施例1の文字処理装置における文字パターンを生成するための処理手順を表すフローチャートである。

【図14】実施例2における文字処理装置のRAMのデータ構成を表す図である。

【図15】書体別インデックステーブルのデータ構成を表す図である。

【図16】文字コード別インデックステーブルのデータ構成を表す図である。

【図17】実施例2の文字処理装置における登録処理の処理手順を表すフローチャートである。

【図18】本実施例2の文字処理装置における文字パターンを取得するための処理手順を表すフローチャートである。

10 【図19】実施例3の相対テーブルのデータ構成を表す図である。

【図20】実施例3における相対テーブルと各書体ファイル間の関連を表す図である。

【図21】実施例3における書体別インデックステーブルのデータ構成を表す図である。

【図22】実施例3における文字コード別インデックステーブルのデータ構成を表す図である。

【図23】実施例3における代用文字別インデックステーブルのデータ構成を表す図である。

20 【図24】実施例3の文字処理装置における外字登録の手順を表すフローチャートである。

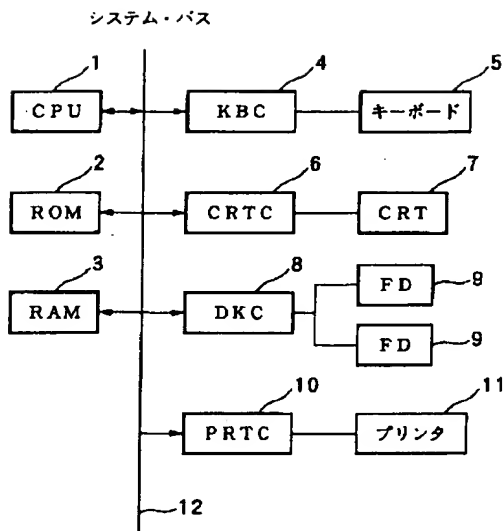
【図25】実施例3の文字処理装置における外字登録の手順を表すフローチャートである。

【図26】実施例4の文字処理装置の外字取得手順を表すフローチャートである。

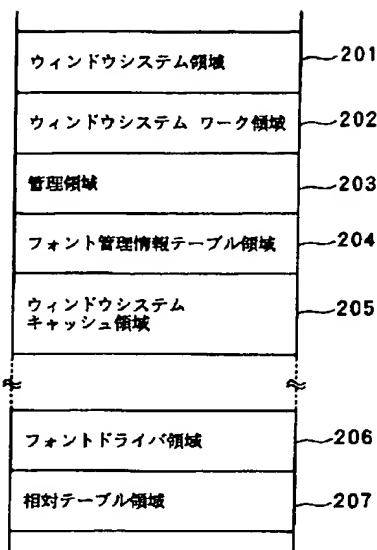
【符号の説明】

- 1 CPU
- 2 ROM
- 3 RAM
- 30 4 キーボード制御部 (KBC)
- 5 キーボード
- 6 ディスプレイ制御部 (CRTC)
- 7 ディスプレイ装置 (CRT)
- 8 ディスク制御部 (DCK)
- 9 外部記憶装置
- 10 プリンタ制御部 (PRTC)
- 11 プリンタ
- 21 外字ファイル
- 31 明朝体ファイル
- 40 32 ゴシック体ファイル
- 33 毛筆体ファイル
- 41 相対テーブル

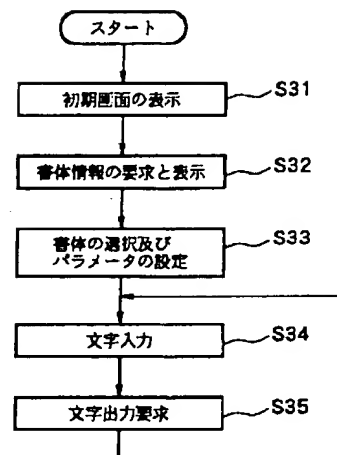
【図 1】



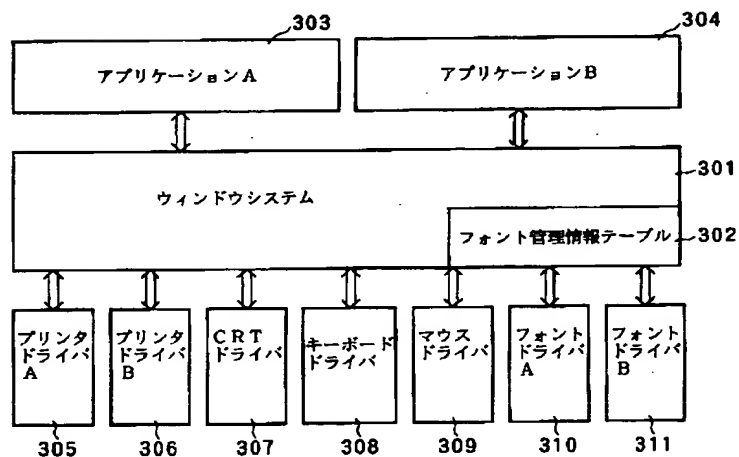
【図 2】



【図 6】



【図 3】

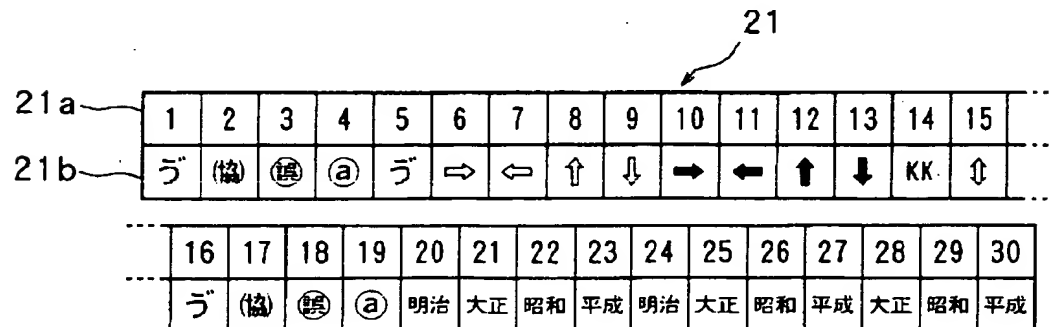


【図 7】

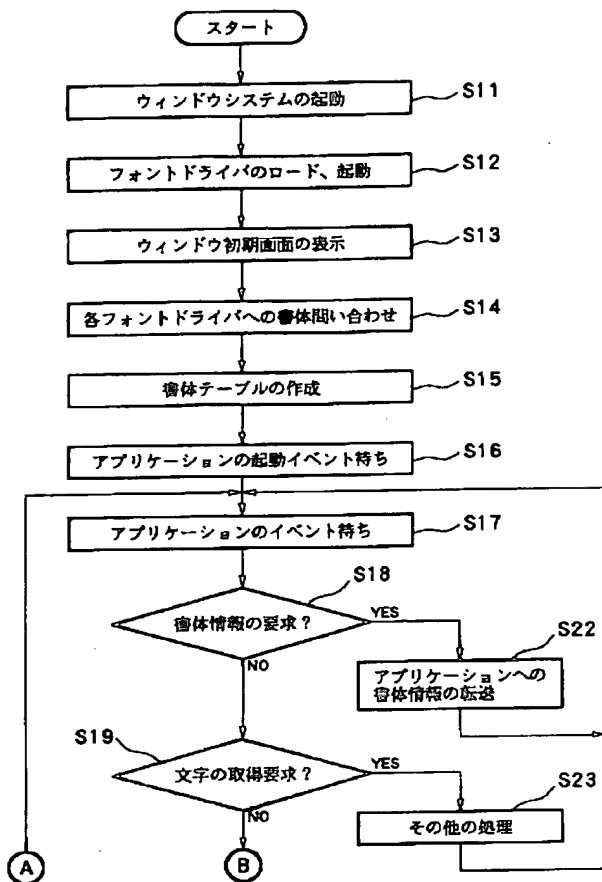
書体リスト

明朝体
ゴシック体
丸ゴシック体1
丸ゴシック体2
毛筆体
外字書体

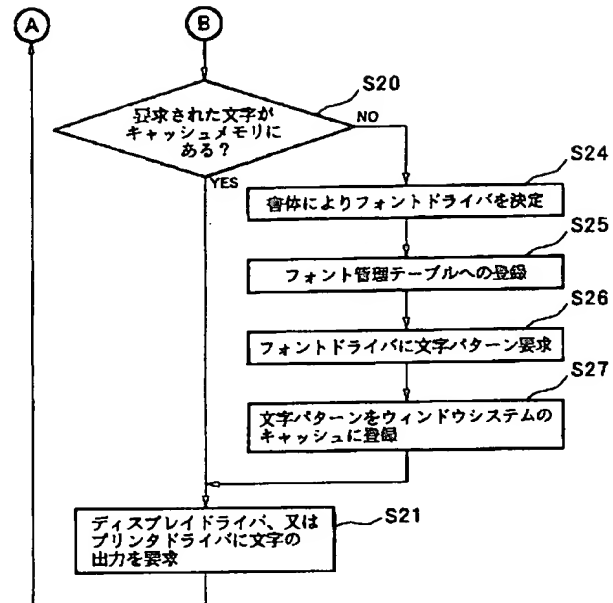
【図 8】



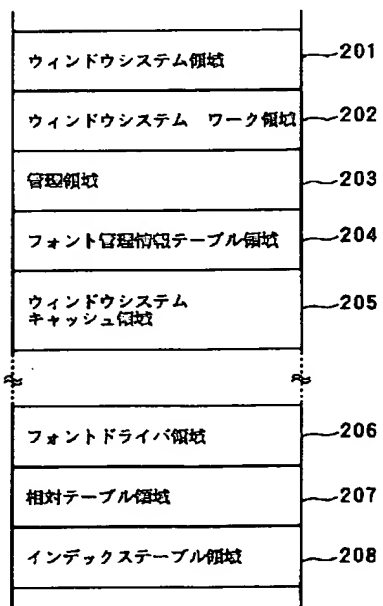
【図 4】



【図 5】



【図 1 4】



明朝体

31a

31

2421	2422	2423	2424	2425	2426		3021	3022	
あ	あ	い	い	う	う		亜	啞	

31b

3023	3024	3025	3026	3027	3028	3029	302A	
娃	阿	哀	愛	挨	始	逢	葵	

ゴシック体

32

2421	2422	2423	2424	2425	2426		3021	3022	
あ	あ	い	い	う	う		亜	啞	

3023	3024	3025	3026	3027	3028	3029	302A	
娃	阿	哀	愛	挨	始	逢	葵	

毛筆体

33

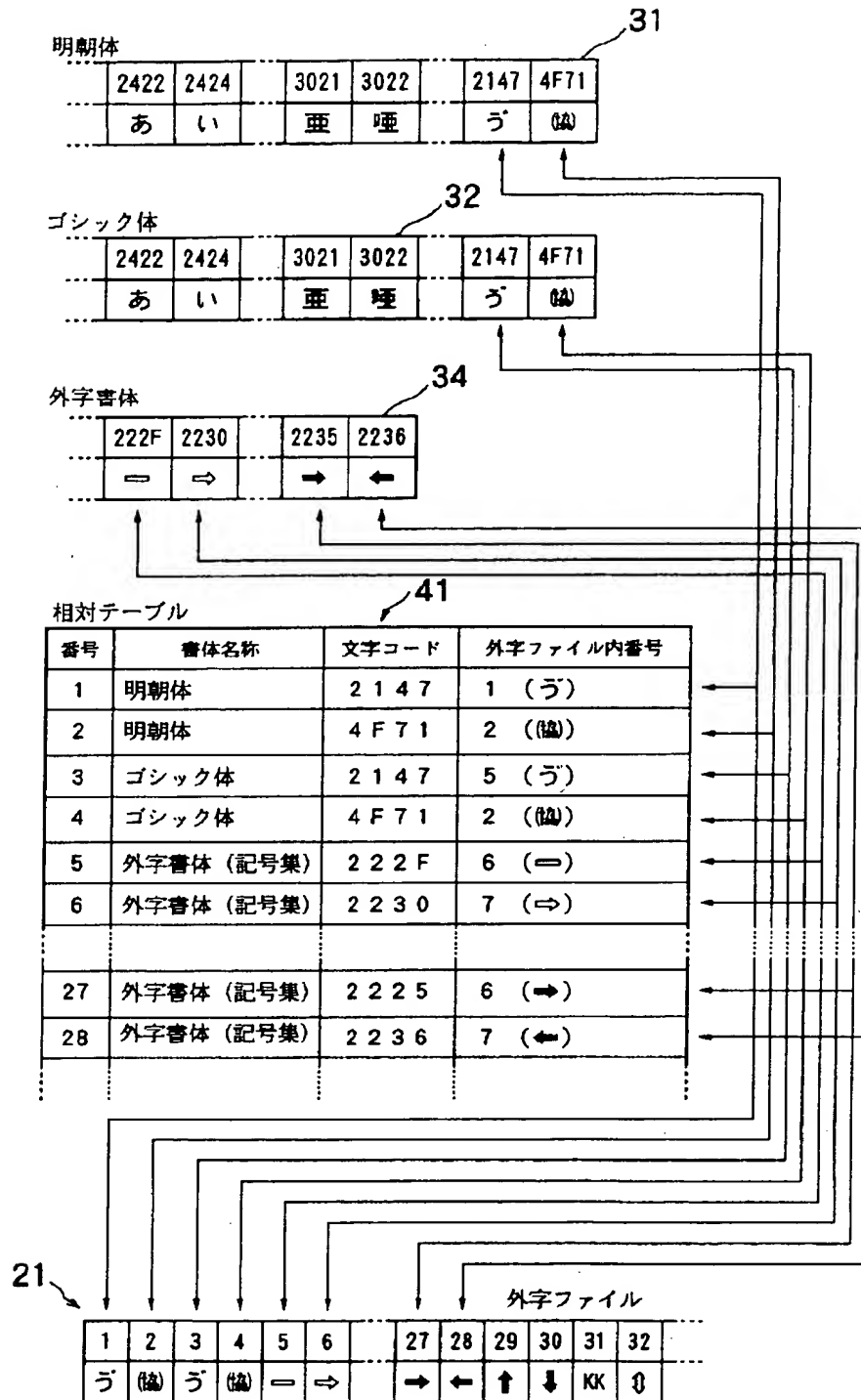
2421	2422	2423	2424	2425	2426		3021	3022	
あ	あ	い	い	う	う		亜	啞	

3023	3024	3025	3026	3027	3028	3029	302A	
娃	阿	哀	愛	挨	始	逢	葵	

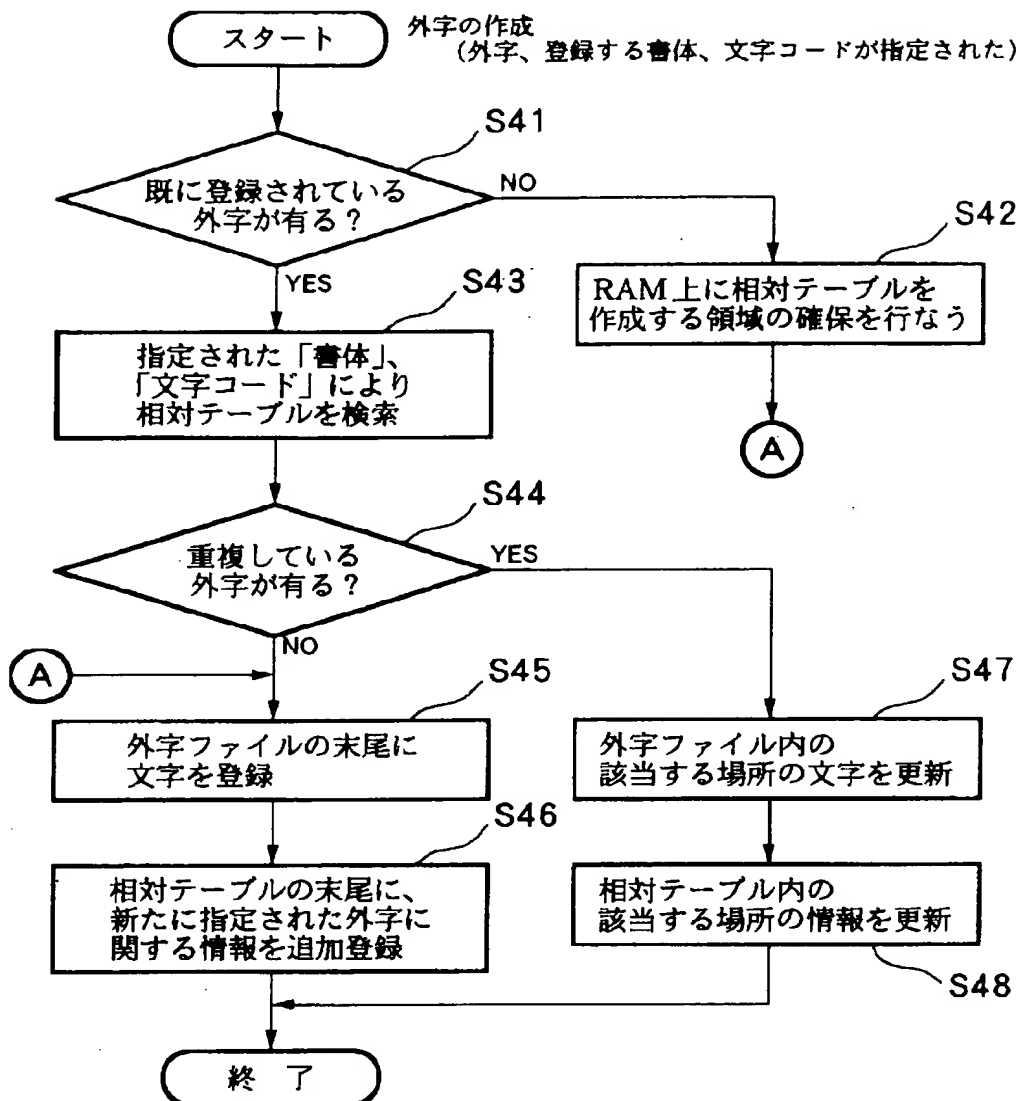
【図 1 0】

41a 番号	41b 書体名称	41c 文字コード	41d 外字ファイル内番号
1	明朝体	2 1 4 7	1 (う)
2	明朝体	4 F 7 1	2 (協)
3	明朝体	4 F 6 C	3 (調)
4	明朝体	2 6 6 1	4 (a)
5	ゴシック体	2 1 4 7	5 (う)
6	外字書体 (記号集)	2 2 2 F	6 (一)
7	外字書体 (記号集)	2 2 3 0	7 (⇒)
16	毛筆体	2 1 4 7	16 (う)
17	毛筆体	4 F 7 1	17 (協)
18	毛筆体	4 F 6 C	18 (調)
20	明朝体	2 D 6 D	20 (明治)
21	明朝体	2 D 6 E	21 (大正)
22	明朝体	2 D 6 F	22 (昭和)
23	明朝体	2 E 3 F	23 (平成)
24	ゴシック体	2 D 6 D	24 (明治)
25	ゴシック体	2 D 6 E	25 (大正)
26	ゴシック体	2 D 6 F	26 (昭和)
27	ゴシック体	2 E 3 F	27 (平成)
28	毛筆体	2 D 6 E	28 (大正)
29	毛筆体	2 D 6 F	29 (昭和)
30	毛筆体	2 E 3 F	30 (平成)

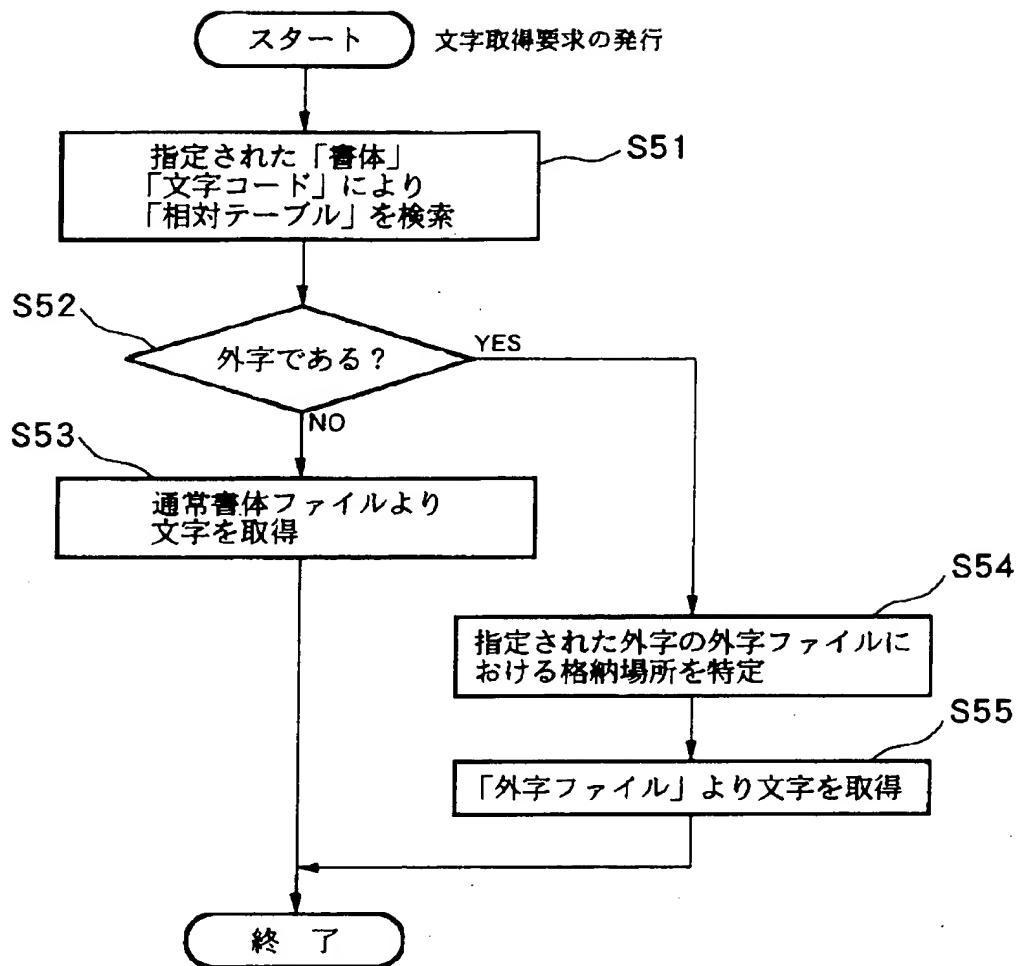
【図11】



【図12】



【図 1 3】



【図 2 3】

番号	文字コード	オリジナル書体名称	代用書体名称	相対テーブル内番号
1	2 6 6 1	明朝体	ゴシック体	4 (㊶)
			毛筆体	4 (㊶)
2	4 F 6 C	明朝体	ゴシック体	3 (㊸)
			毛筆体	3 (㊸)
3	4 F 7 1	明朝体	ゴシック体	2 (㊺)
			毛筆体	2 (㊺)

【図 15】

(A)

51

番号	文字コード	相対テーブル内番号
1	2 1 4 7	1 (う)
2	2 6 6 1	4 (a)
3	2 D 6 D	20 (明治)
4	2 D 6 E	21 (大正)
5	2 D 6 F	22 (昭和)
6	2 E 3 F	23 (平成)
7	4 F 6 C	3 (㊤)
8	4 F 7 1	2 (㊦)

(B)

52

番号	文字コード	相対テーブル内番号
1	2 1 4 7	5 (う)
2	2 D 6 D	24 (明治)
3	2 D 6 E	25 (大正)
4	2 D 6 F	26 (昭和)
5	2 E 3 F	27 (平成)

(C)

53

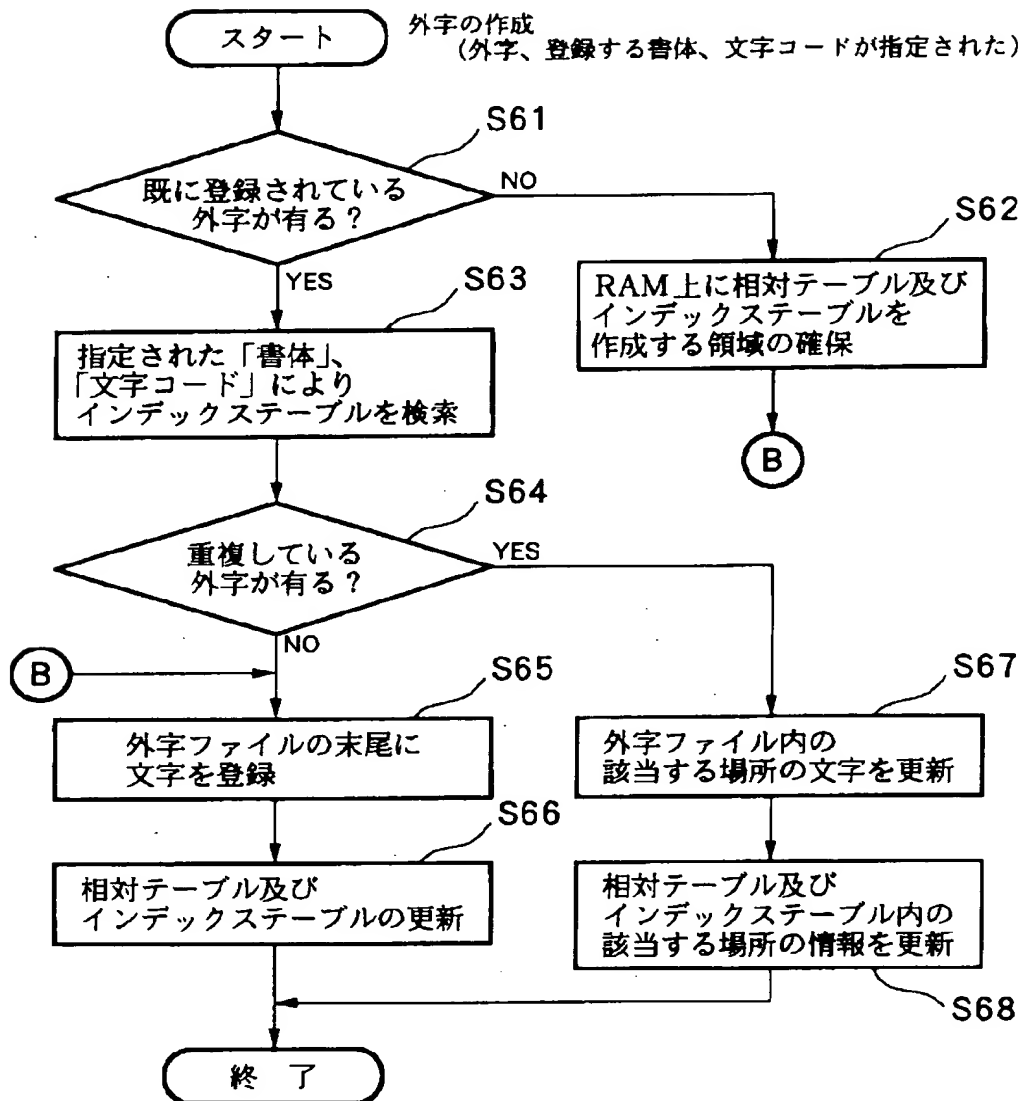
番号	文字コード	相対テーブル内番号
1	2 2 2 F	6 (㊥)
2	2 2 3 0	7 (㊦)

【図 1 6】

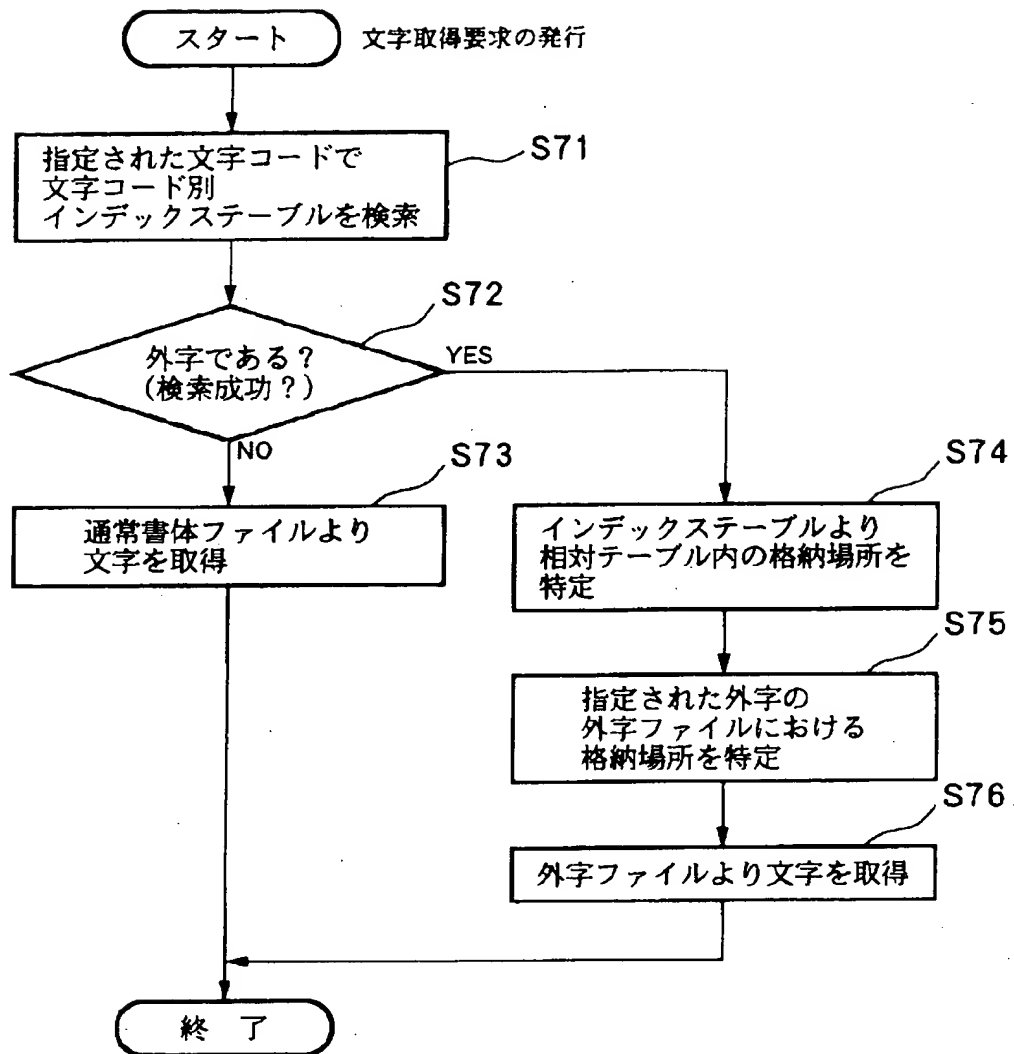
54

番号	文字コード	書体名称	相対テーブル内番号
1	2 1 4 7	明朝体	1 (う)
		ゴシック体	5 (う)
		毛筆書体	16 (う)
2	2 2 2 F	外字書体 (記号集)	6 (＝)
3	2 2 3 0	外字書体 (記号集)	7 (⇒)
n	2 D 6 D	明朝体	20 (明治)
		ゴシック体	24 (明治)
n+1	2 D 6 E	明朝体	21 (大正)
		ゴシック体	25 (大正)
		毛筆書体	28 (大正)
n+2	2 D 6 F	明朝体	23 (昭和)
		ゴシック体	26 (昭和)
		毛筆書体	29 (昭和)

【図 17】



【図 1 8】



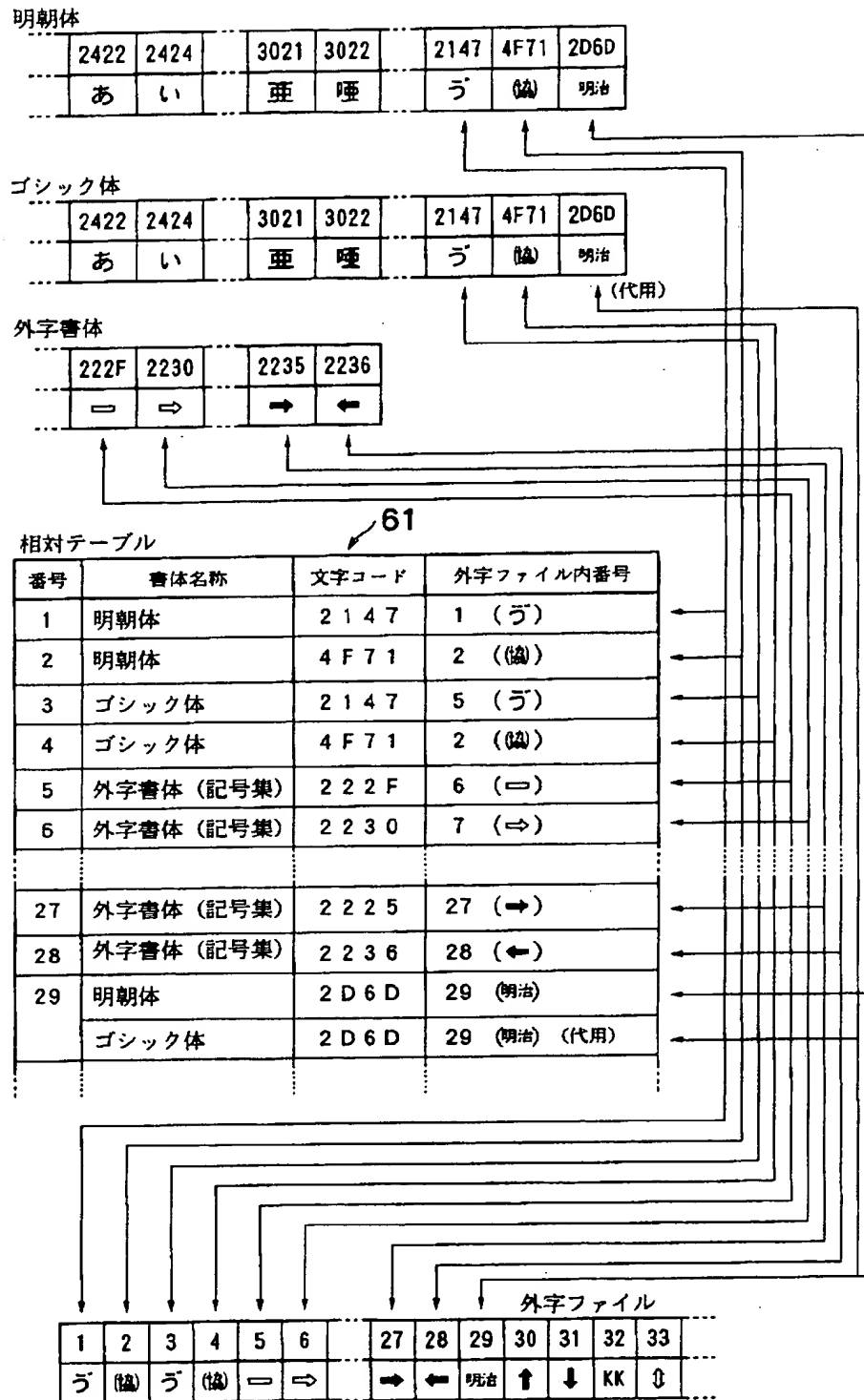
【図 1 9】

61

番号	書体名称	文字コード	外字ファイル内番号
1	明朝体	2 1 4 7	1 (う)
2	明朝体	4 F 7 1	2 (協)
	ゴシック体	4 F 7 1	2 (協) (代用)
3	明朝体	4 F 6 C	3 (調)
	ゴシック体	4 F 6 C	3 (調) (代用)
4	明朝体	2 6 6 1	4 (a)
	ゴシック体	2 6 6 1	4 (a) (代用)
	毛筆体	2 6 6 1	4 (a) (代用)
5	ゴシック体	2 1 4 7	5 (う)
6	外字書体 (記号集)	2 2 2 F	6 (一)
7	外字書体 (記号集)	2 2 3 0	7 (⇒)
16	毛筆体	2 1 4 7	16 (う)
17	毛筆体	4 F 7 1	17 (協)
18	毛筆体	4 F 6 C	18 (調)
20	明朝体	2 D 6 D	20 (明治)
	毛筆体	2 D 6 D	20 (明治) (代用)
21	明朝体	2 D 6 E	21 (大正)
22	明朝体	2 D 6 F	22 (昭和)
23	明朝体	2 E 3 F	23 (平成)
24	ゴシック体	2 D 6 D	24 (明治)
25	ゴシック体	2 D 6 E	25 (大正)
26	ゴシック体	2 D 6 F	26 (昭和)
27	ゴシック体	2 D 3 F	27 (平成)
28	毛筆体	2 D 6 E	28 (大正)
29	毛筆体	2 D 6 F	29 (昭和)
30	毛筆体	2 E 3 F	30 (平成)

62

【図 20】



【図 2 1】

(A)

番号	文字コード	相対テーブル内番号
1	2 1 4 7	1 (う)
2	2 6 6 1	4 (㊤)
3	2 D 6 D	20 (明治)
4	2 D 6 E	21 (大正)
5	2 D 6 F	22 (昭和)
6	2 E 3 F	23 (平成)
7	4 F 6 C	3 (鯨)
8	4 F 7 1	2 (協)

(B)

番号	文字コード	相対テーブル内番号
1	2 1 4 7	5 (う)
2	2 6 6 1	4 (㊤) (代用)
3	2 D 6 D	24 (明治)
4	2 D 6 E	25 (大正)
5	2 D 6 F	26 (昭和)
6	2 E 3 F	27 (平成)
7	4 F 6 C	3 (鯨) (代用)
8	4 F 7 1	2 (協) (代用)

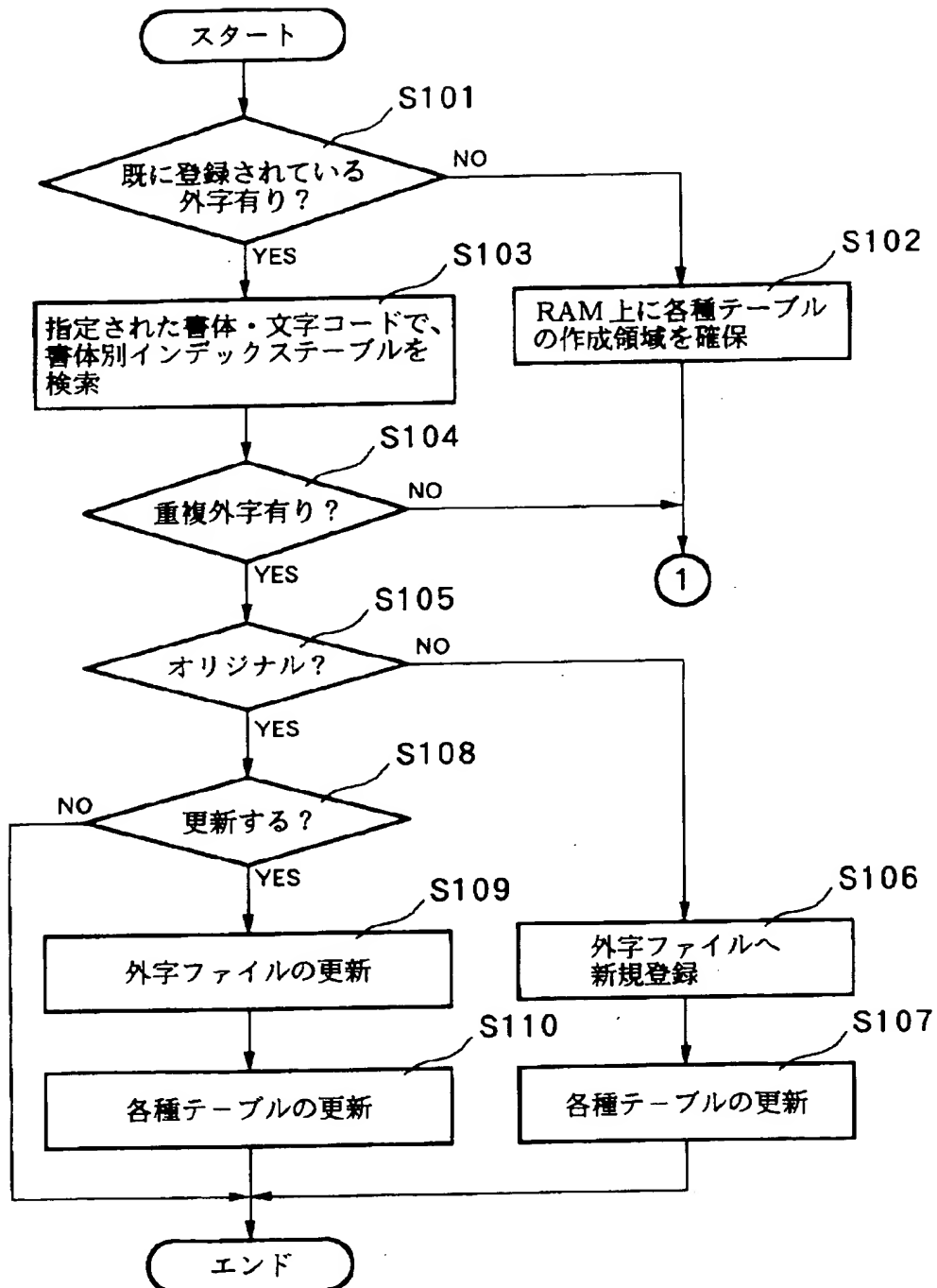
(C)

番号	文字コード	相対テーブル内番号
1	2 2 2 F	6 (⇒)
2	2 2 3 0	7 (⇨)

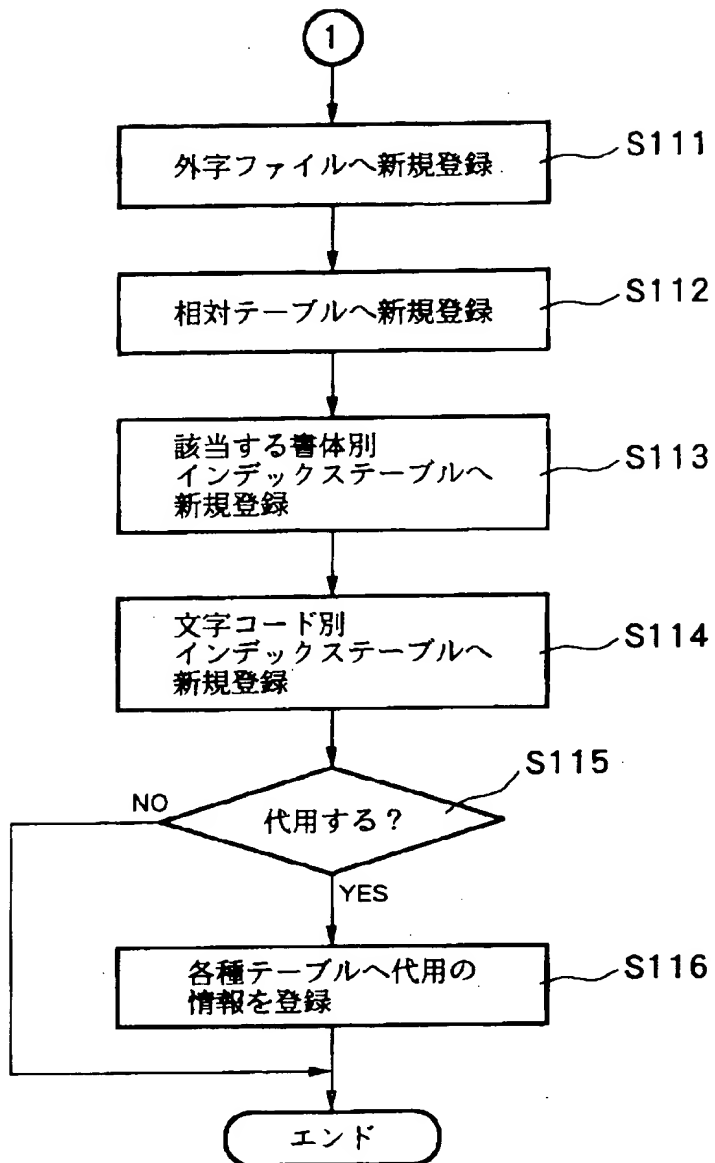
【図22】

番号	文字コード	書体名称	相対テーブル内番号
1	2 1 4 7	明朝体	1 (う)
		ゴシック体	5 (う)
		毛筆書体	16 (う)
2	2 2 2 F	外字書体 (記号集)	6 (⇒)
3	2 2 3 0	外字書体 (記号集)	7 (⇔)
n	2 D 6 D	明朝体	20 (明治)
		ゴシック体	24 (明治)
		毛筆書体	20 (明治) (代用)
n+1	2 D 6 E	明朝体	21 (大正)
		ゴシック体	25 (大正)
		毛筆書体	28 (大正)
n+2	2 D 6 F	明朝体	23 (昭和)
		ゴシック体	26 (昭和)
		毛筆書体	29 (昭和)

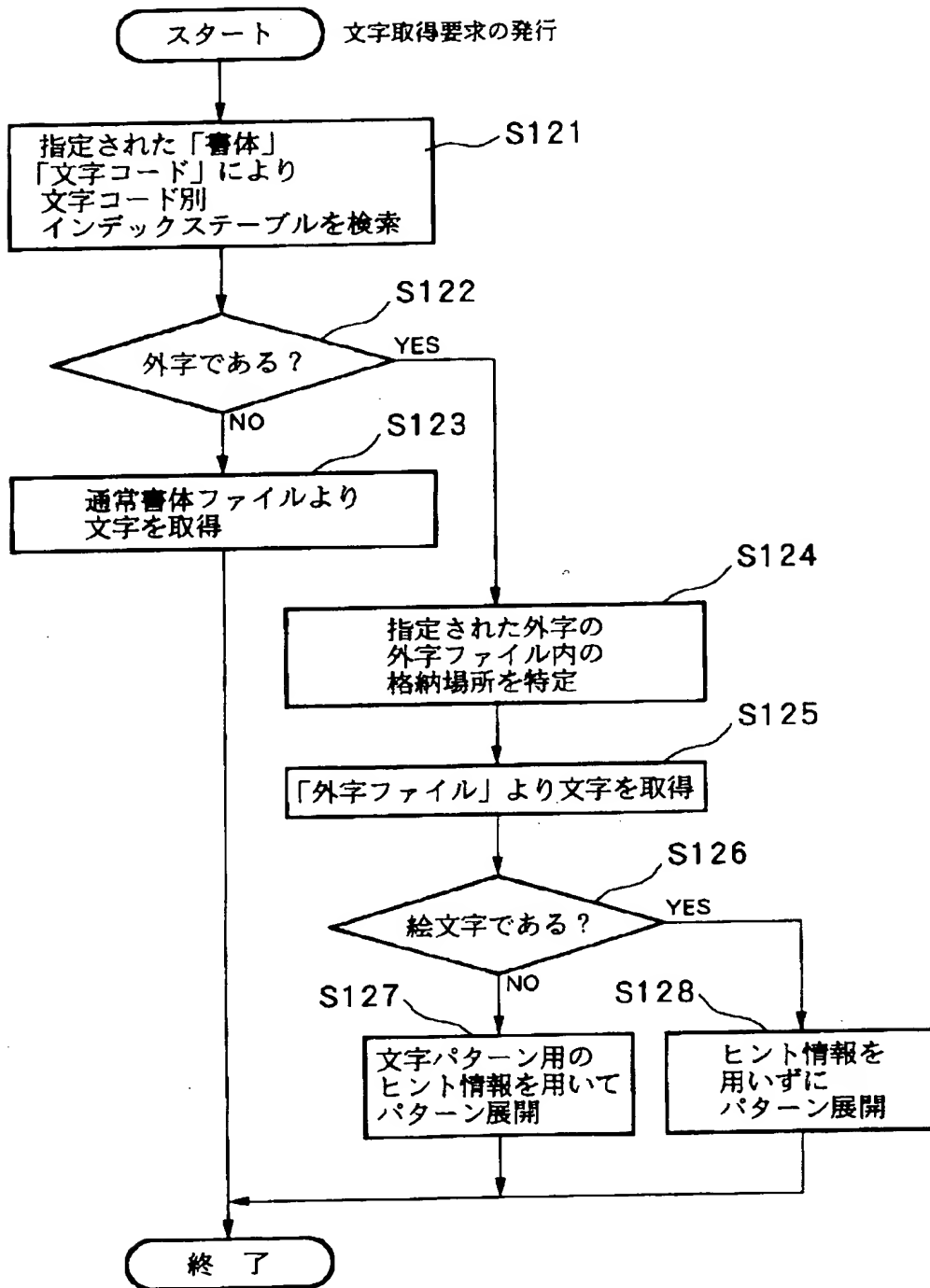
【図24】



【図 2 5】



【図 2 6】



フロントページの続き

(72)発明者 柏木 正樹
東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(72)発明者 松木 浩
東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(72)発明者 佐々木 安彦
東京都大田区下丸子 3 丁目30番 2 号 キヤ
ノン株式会社内

(72)発明者 長田 守
東京都大田区下丸子 3 丁目30番 2 号 キヤ
ノン株式会社内

THIS PAGE BLANK (USPTO)